

X1/X1 turboシリーズ テクニカルノウハウ

X1-Techknow

エックスワン・テクノ

X1
シリーズ

X1-turbo
シリーズ

X1-turbo Z

X1/X1turboシリーズ テクニカルノウハウ

X1-Techknow

エックスワン・テクノウ

▶X1 シリーズ

▶X1-turbo シリーズ

▶X1-turbo Z

はじめに

パソコンも、一時期の熱狂的なブームが去ると共に社会に定着しはじめ、ワープロ等のOA機器あるいはゲームなどに利用されて私達の身近な存在となりました。ビジネスユースでは16ビット機が主流になりましたが、パーソナルユースとしてはまだ8ビット機が大きなシェアを占めています。しかしながらファミリーコンピュータの普及は、ゲーム中心に利用をされてきた従来の8ビット機に大きな転換を迫ったといっていでしょう。

このような状況のなかでX1 turboZは、FM音源標準サポート、4096色同時発色というAV(オーディオ・ビジュアル)機能を全面に押しだして発売されました。テレビ画面のデータをパソコンに取り込むことのできるビデオディジタル機能を含めて、従来のゲームパソコンとは一味異なった方向付けがなされていると言えます。

X1からX1 turboへと大幅に改良されたX1シリーズですが、turboZではグラフィックスを中心に更に大幅に仕様が変更されています。そこでX1 turboZのユーザーがその機能を十二分に引き出し、より有効に使うためには、本体の内部や周辺機器について詳しく知ることがポイントになってきます。

本書は次のように2部構成をとっています。第I部は、X1 turboZの新機能をコンピュータグラフィックスの理論の解説とともに理解する構成をとっています。第II部はX1シリーズの本体はもちろんのこと、プリンター、ディスクユニットに至るまで、内部解析情報や活用のノウハウを実戦に役立つようにまとめてあります。

本書がこの新しい可能性を秘めたX1シリーズを活用するための一助となれば幸いです。

1987年7月 ビー・エヌ・エヌ第二企画部

目 次

はじめに	3
目 次	4

I 部 ノウハウ編

第1章 グラフィックスの理論 9

1-1 I/Oポート	9
1-2 V-RAM構成	11
1-3 V-RAMのアドレス	13
1-4 ディスプレイの表示の仕組み	16
1-5 ウィンドウと座標変換	18
1-6 色の表現(RGBとHSV)	21
1-7 デザリング	28

第2章 グラフィックツールの製作 31

2-1 各種描画アルゴリズム	31
2-2 グラフィックサブルーチン	35

第3章 ステレオグラフィックスの理論 43

3-1 3次元表示の方法	43
3-2 3次元立体表示の仕組み	47

第4章 FM音源によるMUSIC機能 51

4-1 光と音	51
4-2 音色とは	51
4-3 音程とは	52
4-4 音量とは	53
4-5 音色エディタの機能	53
●グラフィックサブルーチン一覧	57

II部 テクニカル編

第1章 システム概説	79
1-1 ハードウェア概説	79
1-2 X1シリーズのハードウェア比較	80
1-3 ソフトウェア概説	84
第2章 メモリー構成	85
2-1 メインメモリー	85
2-2 内蔵ROM	86
2-3 I/O制御	89
第3章 HuBASICの内部構造	93
3-1 HuBASICの種類	93
3-2 HuBASICメモリーマップ	93
3-3 プログラムの格納状態	96
3-4 中間言語	97
3-5 変数テーブル	97
3-6 数値の内部表現	101
3-7 機械語サブルーチンとのリンク	102
第4章 画面表示	105
4-1 V-RAM	105
4-2 画面の構成	125
4-3 テキスト画面	138
4-4 特殊画面制御	154
第5章 サブCPU	175
5-1 サブCPU	175
5-2 キー入力	178
5-3 専用モニターTVのコントロール	184
5-4 専用カセットデッキのコントロール	187
5-5 タイマーのコントロール	191
5-6 PPI(8255)	197

第6章 割り込み 201

6-1	割り込み処理の概要	201
6-2	シリアルI/O	205
6-3	DMA	214
6-4	CTC	220
6-5	キー入力	222

第7章 フロッピーディスク 223

7-1	ディスクの物理フォーマット	223
7-2	BASICのファイル管理	224
7-3	フロッピーディスクコントローラ(FDC)	226
7-4	FDDのアクセス	231

第8章 サウンド機能 235

8-1	PSG	235
8-2	FM音源	242

第9章 各種インターフェイス 263

9-1	プリンターインターフェイス	263
9-2	ジョイスティックインターフェイス	266
9-3	マウスインターフェイス	267
9-4	RS-232Cインターフェイス	268
9-5	デジタルテロップ	270
9-6	ビデオマルチプロセッサ	271

付 録

A. I/Oマップ 274

B. turboシリーズ 285

B-1	BIOS-ROMマップ	285
B-1-1	項目別	285
B-1-2	アドレス順	294
B-2	ワークエリア	314

索引	331
----------	-----

I 部 ノウハウ編

第 I 部をお読みにする前に

第 I 部では、X1 シリーズの最新マシンである X1turboZ について、その特徴であるグラフィックス機能を中心に解説してゆきます。各章の内容は次のとおりです。

第 1 章 コンピュータグラフィックスを行う場合に最低限必要とされる事柄について解説します。

第 2 章 X1turboZ に特徴的なグラフィックス機能を、BASIC でグラフィックツールを作ることにより解説します。

第 3 章 X1 シリーズのオプションとして発売された立体映像セット(液晶シャッタ方式の 3 D メガネ)を含めたステレオグラフィックスについての理論を解説します。

第 4 章 FM 音源に関する基本的な事柄について解説します。

第 I 部は基本的には、X1turboZ ユーザーを対象に書かれています。従って、第 I 部で掲載したプログラムは turboZ 以外の機種に関して、動作確認を行っていませんので、turboZ 以外の機種をお持ちの方はご注意ください。

第1章

グラフィックスの理論

コンピュータグラフィックス(Computer Graphics 略してCGと呼ぶ)というと読者の皆さんはどのようなものを思い浮かべるでしょうか。多分、テレビのコマーシャルや映画等で使用されているきらびやかな映像をイメージされることと思われます。

しかし、そのようなきらびやかな映像も、元を問いただせば頭にコンピュータとつく以上、当り前ですがコンピュータによって作られたものなのです。従ってCG理論の基本というのはコンピュータ上でどのようにしてグラフィックス処理を行うかという、これもまた至極当り前の所から始まっているといえます。

本章では、コンピュータとして X1turboZ を使用した場合のコンピュータグラフィックスに関して、最低限知っておかなければならない事柄を解説します。

1-1 I/Oポート

パソコンは、CPU だけですべての処理を行うわけではなく、SIO,CTC などという周辺 LSI にもある程度処理を任せて、処理の高速化を計っています。I/O ポートは、これらの周辺 LSI とやりとりするための窓口のようなものです。パソコンを一つの都市として考えると、CPU は中央指令部で、メモリは住所(アドレス)を持った町と考えられます。そうすると I/O ポートは、港のようなものでしょうか。

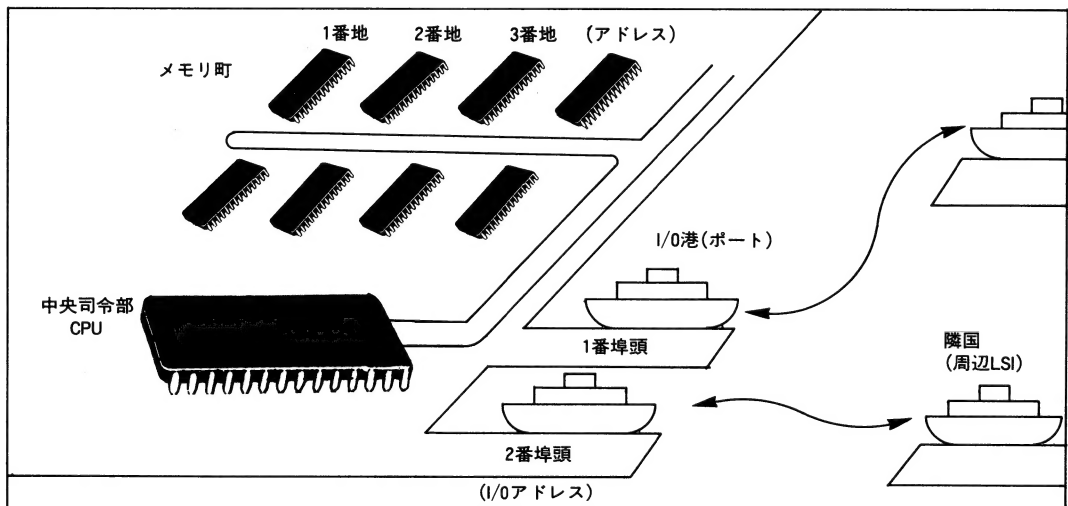


図1-1 I/Oポートは周辺装置とのデータを入出力する港のこと

コンピュータ用語では、アドレス、ポート、マップなど都市にたとえられた表現がよく使われます。これは基板上に立ち並ぶIC群があたかも都市のように見えるためなのかもしれません。

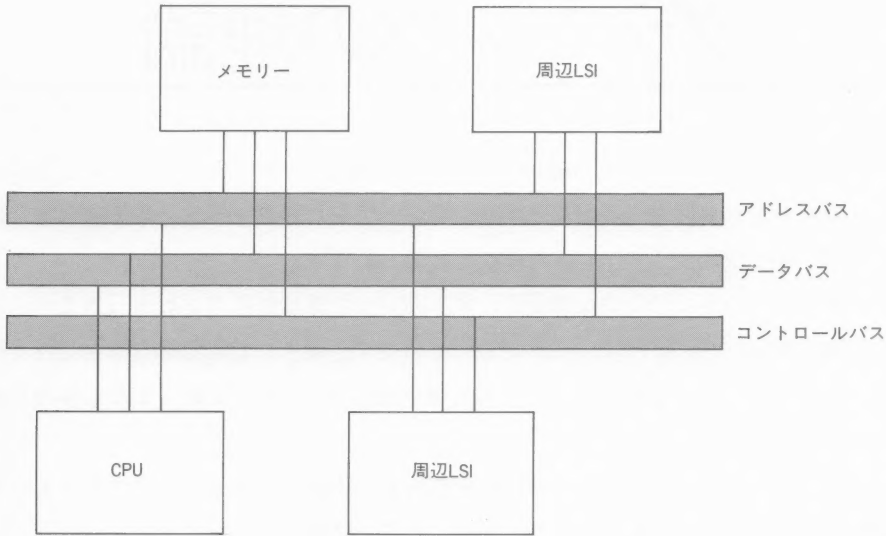


図1-2 パーソナルコンピュータのブロック図(図1-1と同じ意味)

BASIC の OUT &H1FB0,&H80 というコマンドは&H1FB0 で指定される周辺 LSI に対し、&H80 というコマンド又はデータを送るということです。

OUT &H1FB0, &H80
~~~~~  
出力する どこへ    何を  
                    (何番埠頭へ)

逆に A=INP(&H1FB0) は、&H1FB0 で指定される周辺 LSI から、データを読み込み、その読み込んだデータを A に入れるということです。

A    =    INP    (&H1FB0)  
~~~~~  
どこへ 入力する どこから
(何番地へ) (何番埠頭から)

指定された周辺 LSI に対して、データ又はコマンドを送るというと、指定された住所をたよりに郵便屋さんが手紙を届けるイメージに受けとられます。このように考えても全く問題はないのですが、実際に CPU が内部で行っている処理は多少異なっています。CPU は、周辺 LSI に対して、それぞれ別々のコマンドを送っているわけではなく、すべての周辺 LSI に対して同じコマンドを送っているわけです。それを周辺 LSI 側で、自分に対して送られたコマンドかどうか判断し(これをデコードするといいます)自分に対して送られたコマンドのみを実行します。

従って内部処理的にみると、CPU は全ての周辺 LSI にスピーカーで呼びかけていると考えた方が良いでしょう。

先程の例だと、

「&H1FB0 さん、&H80 のコマンドを実行して下さい」

と呼びかけているわけです。

よく I/O マップなどに、&H0F ** という記述がなされていることがありますが、これは下位 2 バイトはなんでもよいということを意味しています。つまり、I/O ポートを呼び出すとき、フルネームで呼び出さず、名字だけで呼び出すようなものでしょう（一般にこのような場合デコードしていないといえます）。

1-2 V-RAM 構成

画面上に表示されるデータが保存されているメモリーのことを V-RAM と呼びます。V-RAM には、キャラクタコードを保存しているテキスト V-RAM と画面の 1 ドット毎の色情報を保存しているグラフィック V-RAM とがあります。このグラフィック V-RAM も色情報のデータの持ち方によって、パレットモード（コンパチモード）と多色モードに分かれます。この多色モードは、X1turboZ によって拡張されたモードです。

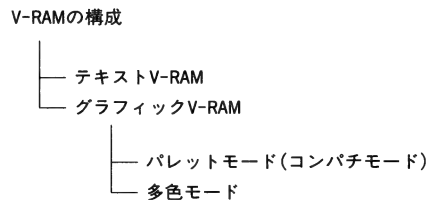


図1-3

※パレットモードのことをマニュアルでは、コンパチモードと呼んでいます。しかし、コンパチモードとここで表記すると、更に新しい機種が出たときに、その機種の多色モードもコンパチモードと呼ばないわけにはいかなくなってしまう混乱する可能性があります。そのため、コンパチモードをパレットモードと表記することにします。

●パレットモードと多色モードについて

パレットモードとは、各画素の色コードを V-RAM 上に持つモードです。色コードとは、パレットレジスタ（ルックアップテーブルともいう）のインデックス値のことであり、X1turboZ では、このインデックスを、アドレスレジスタとデータレジスタに分けて設定します。画面上には、このパレットレジスタにより、R,G,B の各輝度に変換されて表示されます。

これに対して多色モードとは、各画素の輝度を V-RAM 上に持つモードです。即ち、各ピクセル毎に R,G,B それぞれの輝度を直接持つわけです。パレットモードでは色数が V-RAM の枚数により制限されてしまいますが、多色モードの場合このような制限は全くありません。

このように V-RAM の使い方には 2 通りあり、各々特徴的な機能を持っていますが、パレットモードに対して多色モードでは、パレットレジスタを固定的に使用しているだけなので、ハードウェアは、全く共通です。

I 部の最後に、4096色同時表示モードと、64色 2 画面モードのパレットレジスタ初期化プログラムを掲載します。4096色モードから64色 2 画面モードへと切り変えるにはこのパレットレジスタ初期化プログラムを起動させないと正常な画面は表示されません。

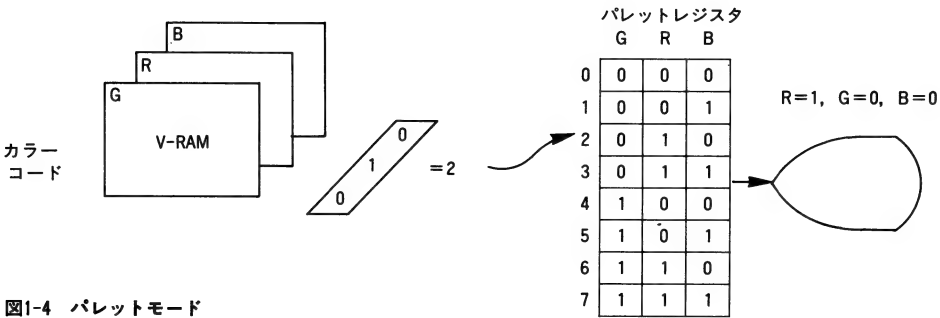


図1-4 パレットモード

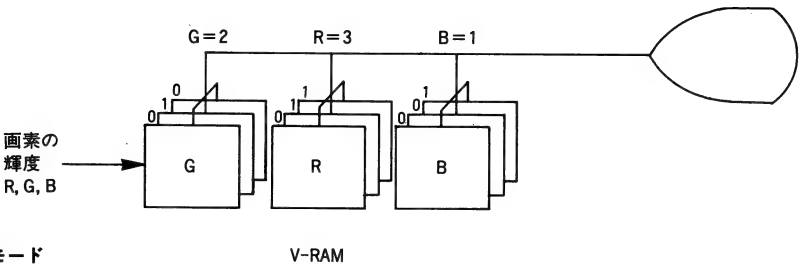


図1-5 多色モード

●パレットモードと多色モードの違い

	パレットモード	多色モード
色数	V-RAMの数により制限あり。 例：3面のときは8色	制限なし
階調数	階調数を増やすと色数が減るため、 階調表現がしにくい。	V-RAMの数 12面のとき RGB 各 4 階調 6面のとき RGB 各 2 階調
パレット機能	有	無
使用法	色コード毎に色をリアルタイムに変更 できるため、色の配色を見るとき に便利。	R, G, Bの輝度を直接持つため、ビ デオ画像の取り込みや、レイトレー シングなどのコンピュータグラフィ ックスに便利

表1-1 パレットモードと多色モードの違い

1-3 V-RAM のアドレス

表示モードは解像度によって数種類あります。しかし、大きく分類すると、縦方向と横方向の解像度により、次表のように分類できます。V-RAM のアドレスは、このモードにより決定されます。

		横方向の解像度	
		WIDTH 40 320ドット	WIDTH 80 640ドット
縦方向の 解像度	200ドット	320×200	640×200
	400ドット 192(384)	320×400 320×192(384)	640×400 640×192(384)

表1-2 ディスプレイの解像度 (X1turboZ)

192ドットモードは、低解像度モニターで高解像度モニターでしか使えない400ドットモードの漢字等を表示するために設けられた互換モードです。従って V-RAM のアドレスは400ドットモードとほぼ同じ構成となっています。

各モードにおける画面左上の部分の V-RAM アドレスを比較したものを次に示します。

I 部の最後に、VRAM のアドレス計算プログラムを掲載してあります。

横 縦	320 200	640 200	320 400 (192/84)	640 400 (192/384)
1	4000	4001	4000	4001
2	4800	4801	4400	4401
3	5000	5001	4800	4801
4	5800	5801	4C00	4C01
5	6000	6001	5000	5001
6	6800	6801	5400	5401
7	7000	7001	5800	5801
8	7800	7801	5C00	5C01
1	4028	4029	6000	6001
2	4828	4829	6400	6401
3	5028	5029	6800	6801
4	5828	5829	6C00	6C01
5	6028	6029	7000	7001
6	6828	6829	7400	7401
7	7028	7029	7800	7801
8	7828	7829	7C00	7C01
			4028	4029
			4428	4429
			4828	4829
			4C28	4C29
			5028	5029
			5428	5429
			5828	5829
			5C28	5C29
			6028	6029
			6428	6429
			6828	6829
			6C28	6C29
			7028	7029
			7428	7429
			7828	7829
			7C28	7C29
			4050	4051
			4450	4451
			4850	4851
			4C50	4C51
			5050	5051
			5450	5451
			5850	5851
			5C50	5C51
			6050	6051
			6450	6451
			6850	6851
			6C50	6C51
			7050	7051
			7450	7451
			7850	7851
			7C50	7C51

表1-3 画面左上のV-RAMアドレスの比較

● V-RAM のアドレスの求め方

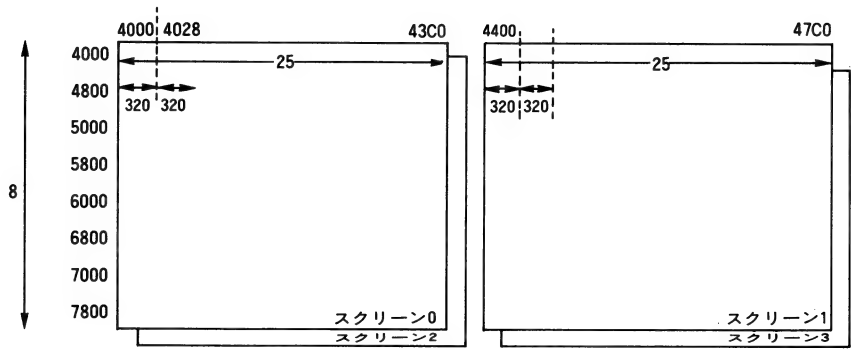


図1-6 320×200ドット(4画面)

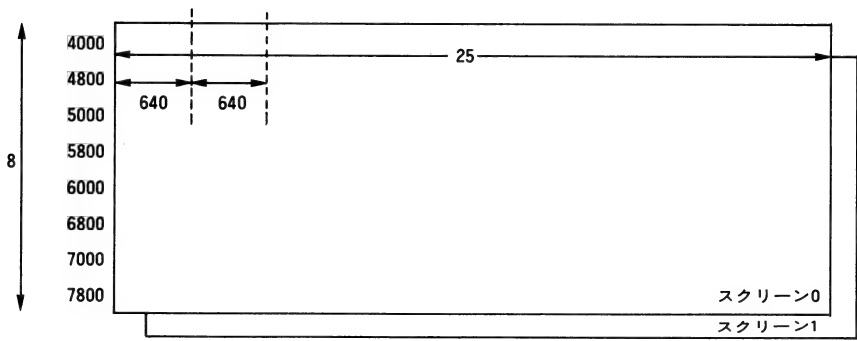


図1-7 640×200ドット(2画面)

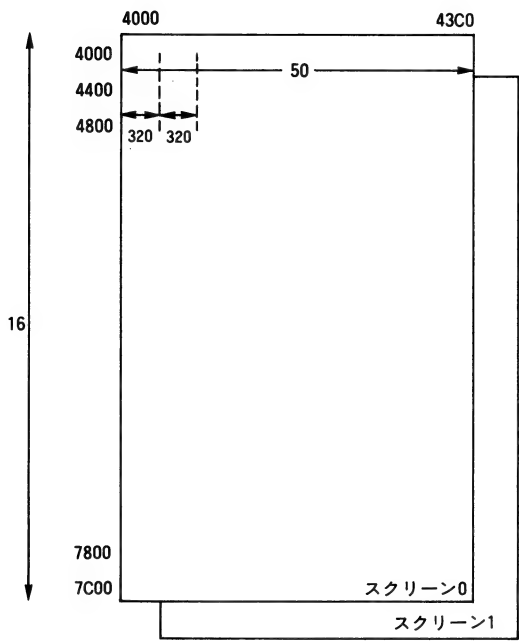


図1-8 320×400(2画面)

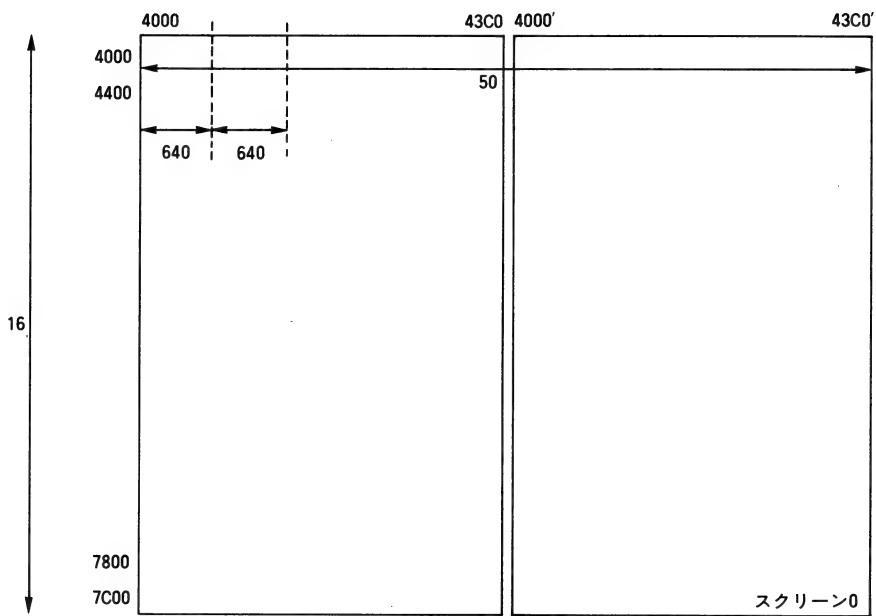


図1-9 640×400(1画面)

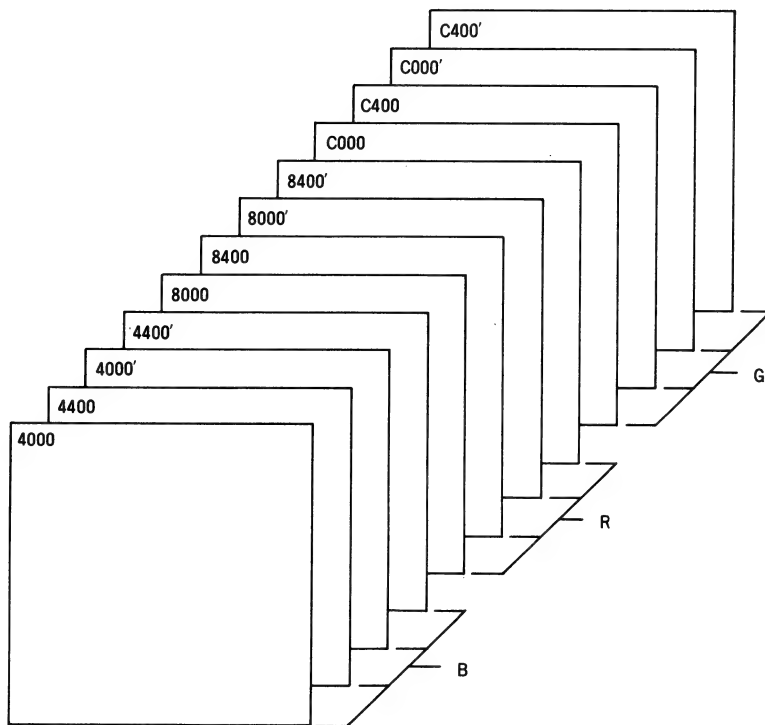


図1-10 320×200(4096色モード)

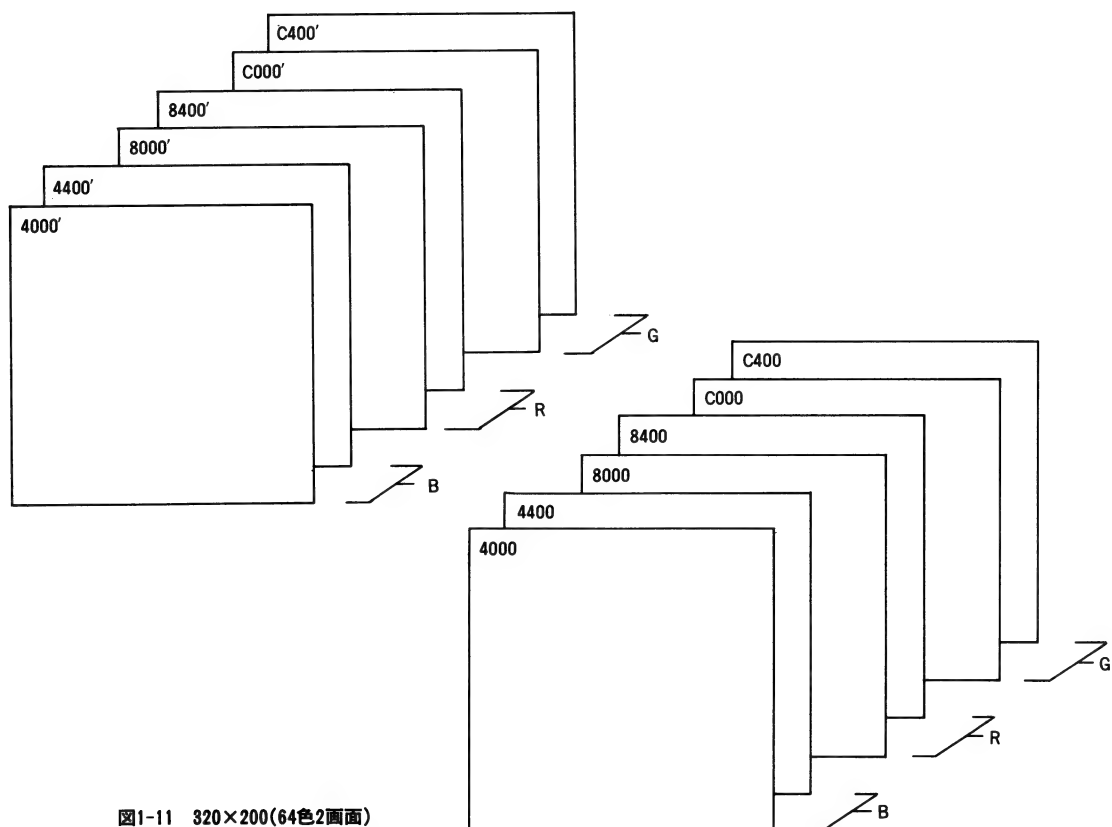


図1-11 320×200(64色2画面)

1-4 ディスプレイの表示の仕組み

ここでテレビ信号の仕様について説明します。

テレビ画面は、RGBの蛍光体に光を当てることによって表示しています。しかし、ある瞬間を見てみると画面上の1ドットしか光っていません。その動作をスローモーションで見るとよく分かるのですが、結局、画面の左上から順番に1ドットずつ光らせているというわけです。画面がちらつきもなく自然に見えるのは、蛍光体がしばらくの間光り続けているのと人間の目の残像現象とによるためです。

また、NTSC信号の場合、走査速度が遅いため、525本の走査線を1本おきに表示するインターレース方式を採用しています。これに対して、高解像度モードでは、1本ずつ表示しています。このような方式を、ノンインターレースといいます。ノンインターレースは、インターレースに比べてちらつきが少なく、見やすい画面となります。

参考までに、EDTVではノンインターレース方式にすることにより、現在のTV放送の画質の向上を目指しています。

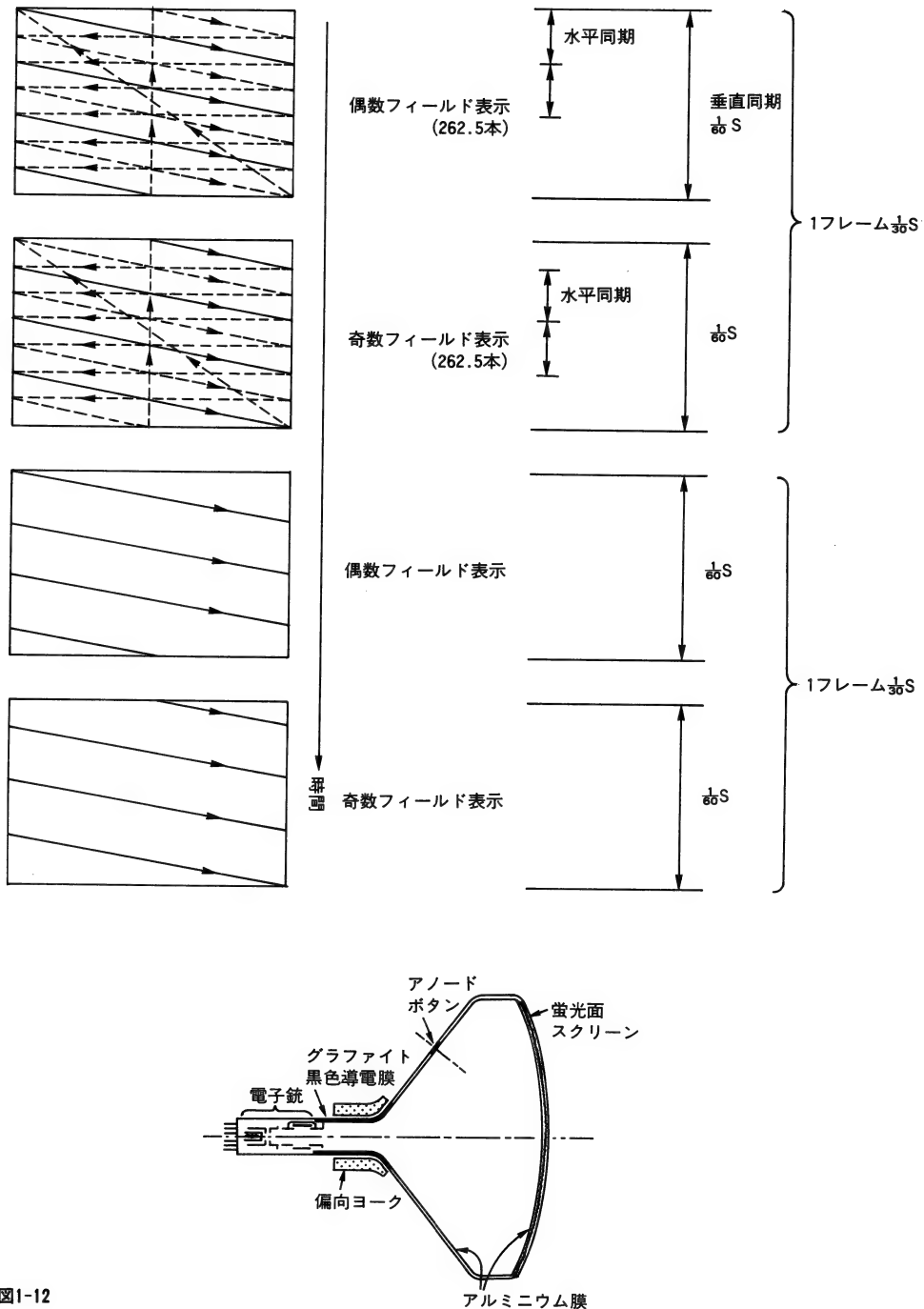


図1-12

テレビの1画面は、偶数フィールドと奇数フィールドから成り立っています。このように、1回の垂直同期の間に表示される262.5本の画像を1フィールドと呼びます。また、偶数フィールドと奇数フィールドを併せたものを、1フレームと呼びます。

1 フィールドを走査するのに1/60秒かかるため1フレームの走査では1/30秒かかります。
従って、1秒間に30枚の画面を見ていることになります。

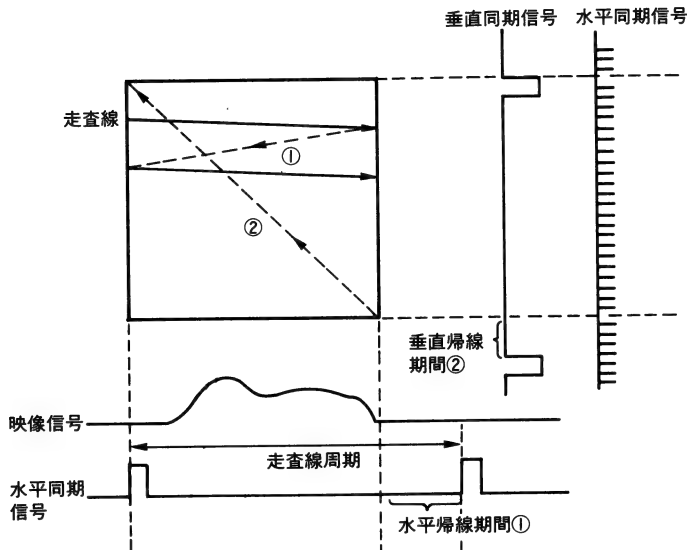


図1-13

1-5 ウィンドウと座標変換

ディスプレイの表示解像度は、使用しているパソコン、及びディスプレイの機種によって異なるのは当然ですが、同一パソコンで複数の表示モードをサポートしているものもあります。この場合、使用するソフトウェアが必要とする色数、解像度に表示を合わせて使用することができます。日本語ワードプロセッサや、表計算ソフトウェアのように色数が少なくても解像度が高いことを要求されるソフトウェアにはそのように合わせ、グラフィックツールやレイトレーシングを行うソフトウェアのように解像度は少し落ちてでも色数が多いことが要求されるソフトウェアにはそのように合わせるという具合に、タイプによって解像度を切り換えるという使い方ができます。

X1turboZでも、

640×400ドット	320×400ドット
640×384ドット	320×384ドット
640×200ドット	320×200ドット
640×192ドット	320×192ドット

の各表示モードがサポートされています。

この表示モードのように画面の1ドットに対応する座標系を、画面座標系又は、デバイス座標系と呼びます。この座標系は、物理的に決定されます。

しかし、アプリケーションソフトウェアを作成するとき、このような物理的な座標系を意識して作ると、汎用性がなくなり、移植性も悪くなってしまいます。

そこで、ユーザー座標系と呼ばれる論理的な座標系を定義できるようになっています。ユーザー座標系は、ワールド座標系ともよばれる実数の範囲で定義される広い空間です。このユーザー座標系と画面座標系を対応づけることをウィンドウ、ビューポート変換と呼びます。これはBASICのWINDOWコマンドに対応します。

$$\text{WINDOW } \underbrace{(X_0, Y_0) - (X_1, Y_1)}_{\text{この値をビューポートと呼ぶ}} - \underbrace{(X_2, Y_2)}_{\text{この値をウィンドウと呼ぶ}}$$

X1のBASICにおいては、WINDOWという1つのコマンドとしてインプリメントされていますが、一般的には、ビューポートという概念とウィンドウという概念に分かれますので、ここでは分けて説明します。

ウィンドウとは、ユーザー座標系のどの部分を表示するかを示すものです。言葉の通り、ユーザー座標系の中に窓をあけて、その中だけが表示されるように決めるわけです。

これに対して、ビューポートとは、ウィンドウによって決められた範囲を画面上のどの部分に表示するかを決めるわけです。

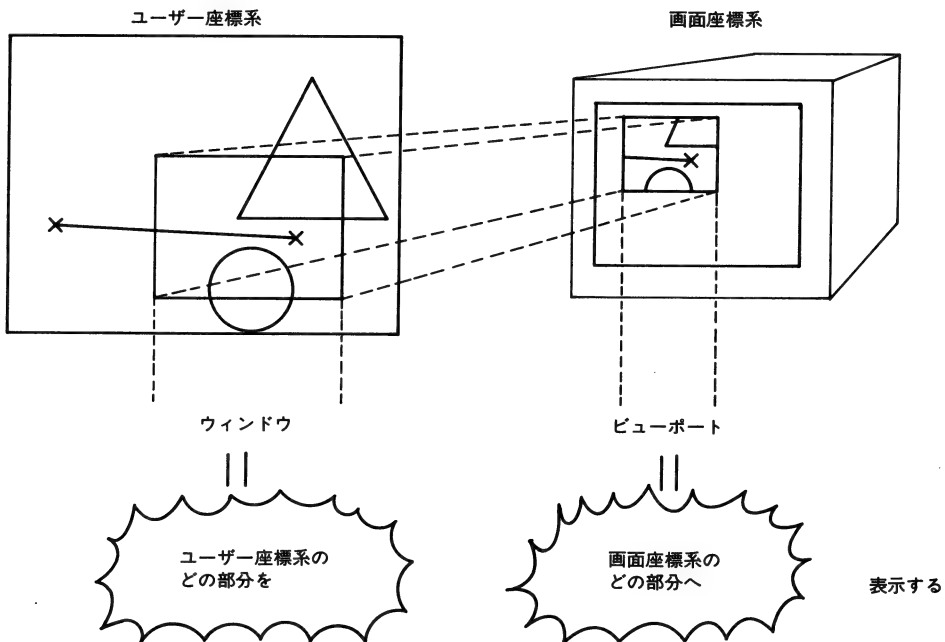


図1-14 ユーザー座標系と画面座標系の対応

これらの、ウィンドウや、ビューポートにより、ユーザー座標系で与えられたデータが、どのように表示されるかを表す変換マトリクスが生成されます。このマトリクスは 3×3 のマトリクスで表され平行移動とスケーリングが行われます。

ここで、ウィンドウ、ビューポート変換マトリクスについて説明します。

ビューポート (Xs, Ys) – (Xe, Ye)

ウィンドウ (X1, Y1) – (X2, Y2)

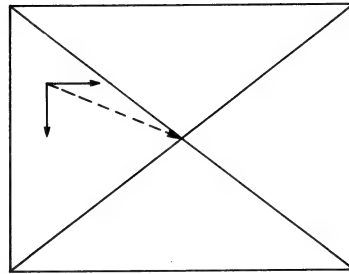


図1-15 平行移動(ウィンドウの中心へ移動する)

まず、ユーザー座標系の原点を、ウィンドウの中心へ移動する変換マトリクスを求めます。

$$M_1 = \begin{bmatrix} 1 & , & 0 & , & 0 \\ 0 & , & 1 & , & 0 \\ -\frac{X_1+X_2}{2} & , & -\frac{Y_1+Y_2}{2} & , & 1 \end{bmatrix}$$

次に、ウィンドウとビューポートの大きさを合わせます。

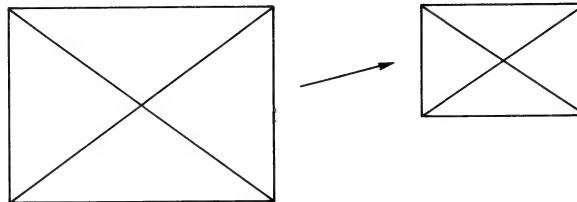


図1-16 スケール(大きさを変える)

$$M_2 = \begin{bmatrix} \frac{X_e-X_s}{X_2-X_1} & , & 0 & , & 0 \\ 0 & , & \frac{Y_e-Y_s}{Y_2-Y_1} & , & 0 \\ 0 & , & 0 & , & 1 \end{bmatrix}$$

最後にビューポートを画面上に配置する変換マトリクスを求めます。

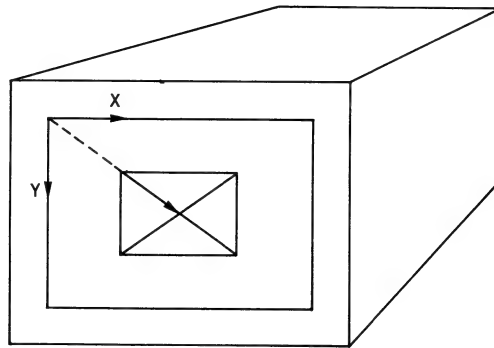


図1-17 平行移動(画面上の表示位置へ移動する)

$$M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{X_s + X_o}{2} & \frac{Y_s + Y_o}{2} & 1 \end{bmatrix}$$

従って、点(X, Y)を変換するとき M1,M2,M3 を掛けあわせればよいので、変換後の点(X',Y')は

$$(X', Y', 1) = (X, Y, 1) \cdot M_1 \cdot M_2 \cdot M_3$$

となります。

この変換のプログラムをI部の最後に掲載してあります。

1-6 色の表現(RGB と HSV)

(1) 色の表現方法

色の表現方法にはいろいろありますが、代表的なものとして以下の方法があります。

- ① RGB モデル パーソナルコンピュータ, グラフィックディスプレイ
- ② HSV モデル 美術, デザイン
- ③ CMY モデル 印刷, 写真のネガ

パーソナルコンピュータは、RGB モデルであり V-RAM や、パレットレジスタの設定は、RGB で行われます。しかし、美術や、デザインの世界では使いにくいので、HSV 方式がよく使われます。CMY モデルは、特殊な用途で使われるため、説明は省略します。

(2) RGB モデルについて

RGB モデルは、直交座標系で表され、Red,Green,Blue の3色の明るさを x,y,z 軸とします。以下に図で示します。

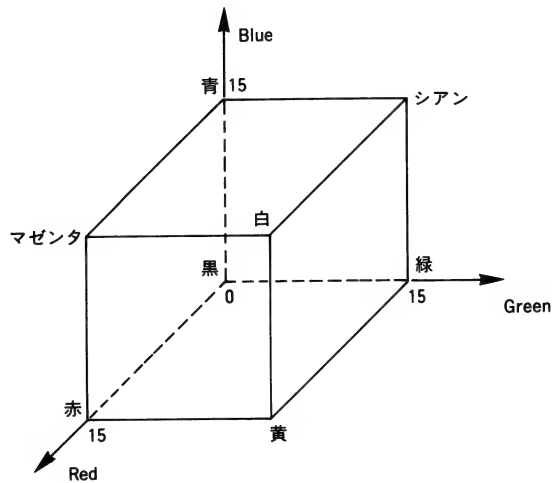


図1-18 RGBモデル

X1turboZ は, R,G,B,各 4 ビットの階調を持つので $2^4 \times 3 = 4096$ 色を表現することができます。

(3) HSV モデル

HSV モデルは, 色相(HUE), 彩度(SATURATION), 明度(VALUE)の 3 つの要素で表されます。

色相とは, 色あいとも呼ばれるもので, 赤青黄などの色の種類を表します。これは, 光のスペクトルにより変化します。ただし, 波長によるスペクトルではなく仮想的なスペクトルで, 色度図と呼ばれます。

彩度とは, 飽和度, あざやかさと呼ばれるもので色の濃淡(純粋さ)を表します。これは, 無彩色(白, 灰, 黒)の割り合いにより変化します。

明度とは, 輝度, コントラストと呼ばれるもので, 色の明るさを表します。これは, 光の強さにより変化します。

これらの 3 つの要素を, 色立体で表すと図1-19のようになります。中心軸の下から上にいくにつれて明度が増加し, また, この軸を中心にして回転する円周上に色相が表され, 角度として表されます。またこのときの半径が彩度を表し, 半径が大きいほど純度が増します。

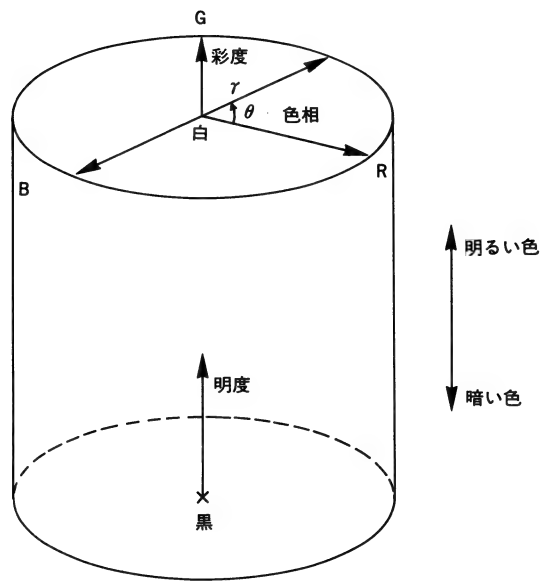


図1-19 HSVモデル

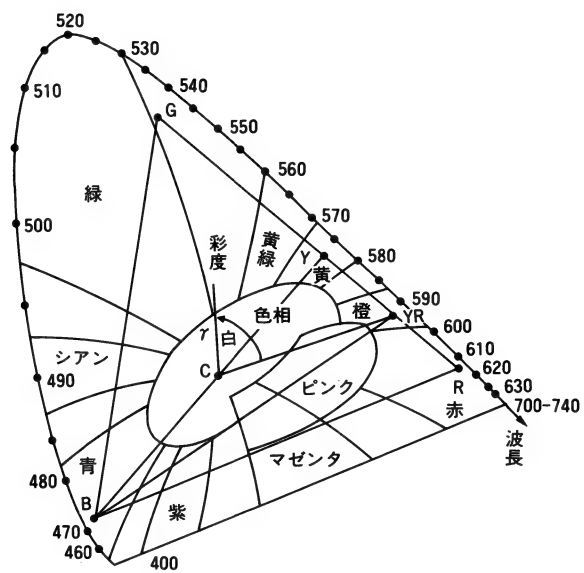


図1-20 色度図

各色の要素とスペクトルと輝度レベルの関係を図で示します。

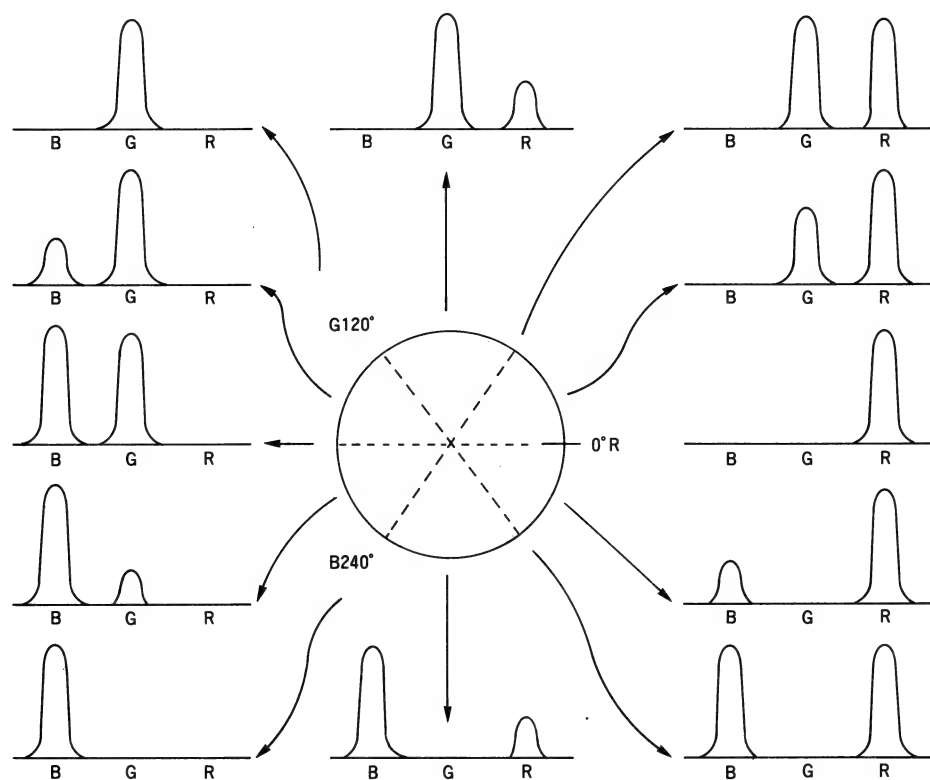
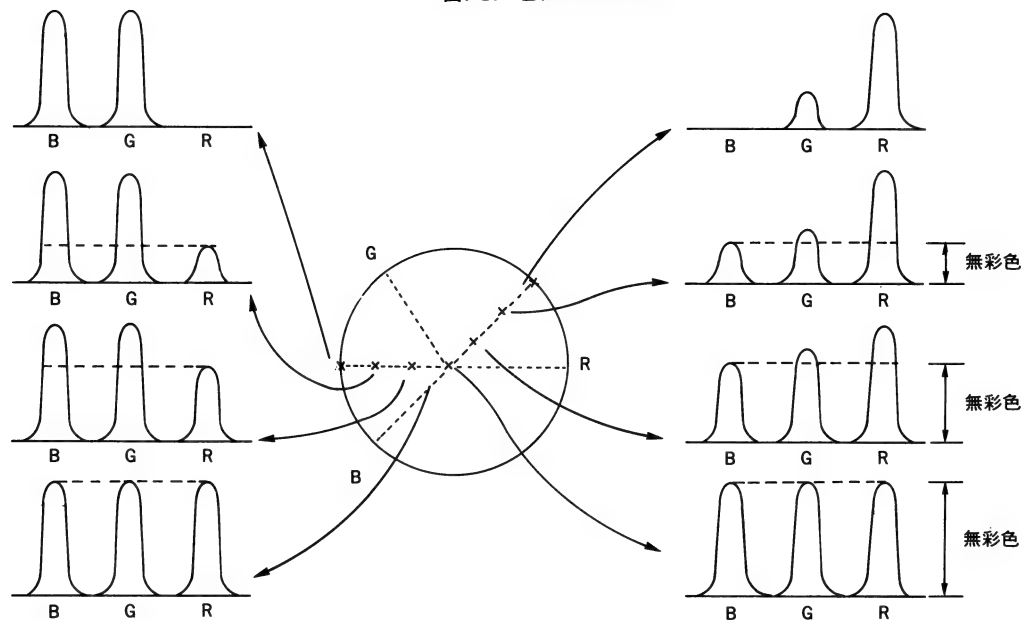


図1-21 色相とスペクトル



R, G, Bが合成されると、白色となり無彩色となる。
彩度とは、無彩色でない部分の比率である。

図1-22 彩度とスペクトル

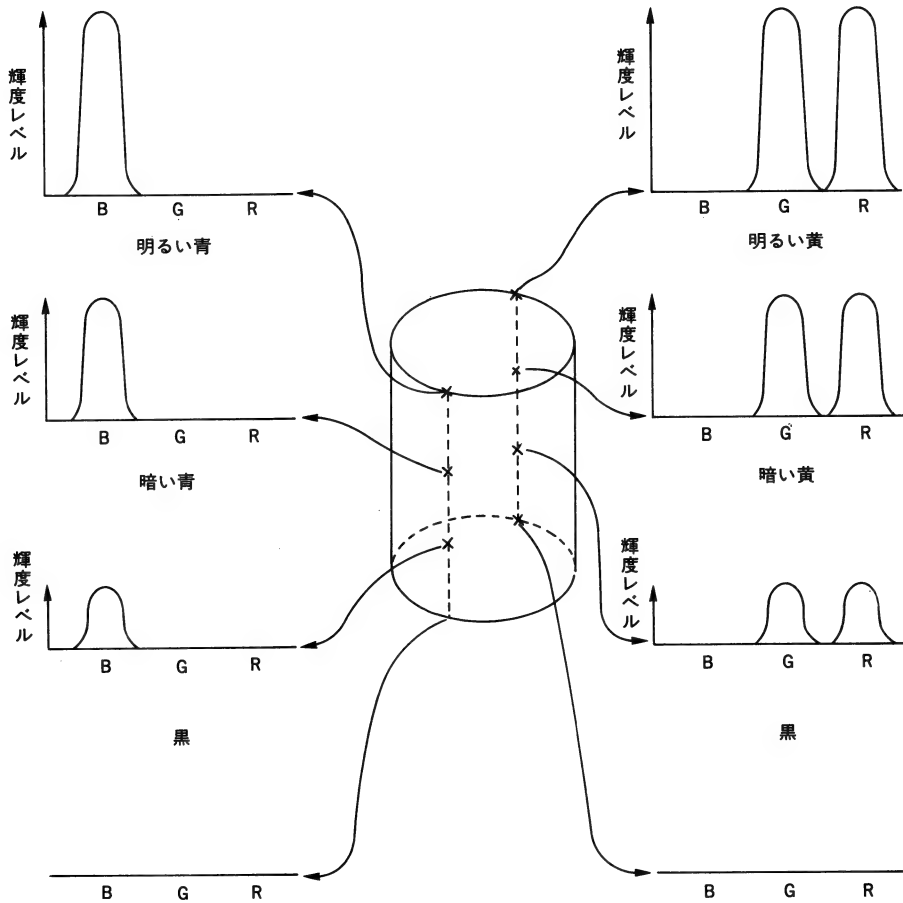


図1-23 明度とスペクトル

(4) HSV モデルから RGB モデルへの変換と逆変換

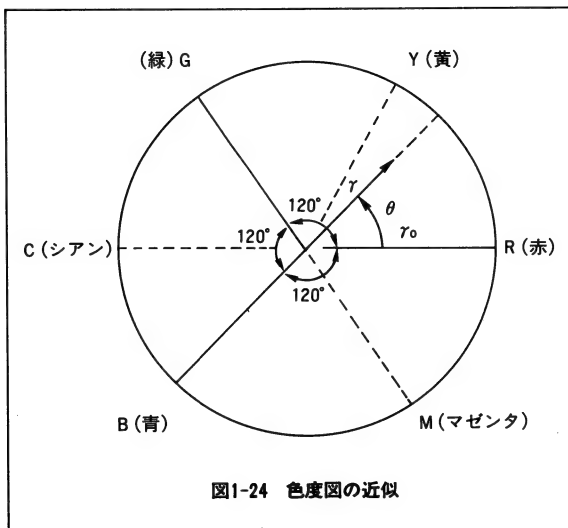


図1-24 色度図の近似

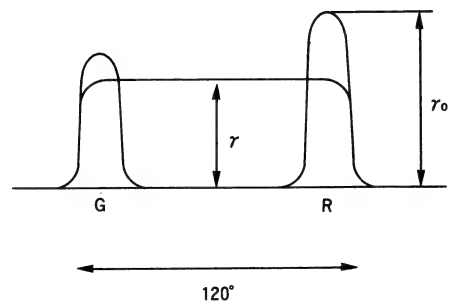


図1-25 スペクトル図

図1-20の色度図を円柱で近似すると図1-24のようになります。

赤を0°とすると、緑は120°、青は240°となります。色相は、度で表す方法とラジアンで表す方法がありますが、度の方が分かりやすいので、度で表すことにします。また彩度と明度は0～100%で表すこととします。

RGB、それぞれの階調数をn、色相をHとすると

$$0^\circ \leq H \leq 60^\circ \quad R_1 = n$$

$$60^\circ \leq H \leq 180^\circ \quad G_1 = n$$

$$180^\circ \leq H \leq 300^\circ \quad B_1 = n$$

$$300^\circ \leq H \leq 360^\circ \quad R_1 = n$$

となります。ここで、彩度Sは $1 - \frac{r}{r_0}$ となるから

$$0^\circ \leq H \leq 120^\circ \quad B_1 = \frac{n \times (100 - S)}{100}$$

$$120^\circ \leq H \leq 240^\circ \quad R_1 = \frac{n \times (100 - S)}{100}$$

$$240^\circ \leq H \leq 360^\circ \quad G_1 = \frac{n \times (100 - S)}{100}$$

となります。残りの1色は、色相の比率により求めることができます。

$$0^\circ \leq H \leq 60^\circ \quad G_1 = n \times \frac{(100 - S)}{100} + n \times \frac{H - 0}{60} \times \frac{S}{100}$$

$$60^\circ \leq H \leq 120^\circ \quad R_1 = n \times \frac{(100 - S)}{100} + n \times \frac{120 - H}{60} \times \frac{S}{100}$$

$$120^\circ \leq H \leq 180^\circ \quad B_1 = n \times \frac{(100 - S)}{100} + n \times \frac{H - 120}{60} \times \frac{S}{100}$$

$$180^\circ \leq H \leq 240^\circ \quad G_1 = n \times \frac{(100 - S)}{100} + n \times \frac{240 - H}{60} \times \frac{S}{100}$$

$$240^\circ \leq H \leq 300^\circ \quad R_1 = n \times \frac{(100 - S)}{100} + n \times \frac{H - 240}{60} \times \frac{S}{100}$$

$$300^\circ \leq H \leq 360^\circ \quad B_1 = n \times \frac{(100 - S)}{100} + n \times \frac{360 - H}{60} \times \frac{S}{100}$$

最後に明度Vを考慮すると

$$R = \frac{R_1 \times V}{100}$$

$$G = \frac{G_1 \times V}{100}$$

$$B = \frac{B_1 \times V}{100}$$

となります。

実際にはこのように場合分けをしなくても、MOD開放などを用いてコンパクトにコーディングできます。

I部の最後に、HSVからRGBへの変換とRGBからHSVへの変換プログラムを掲載します。

(5) 光の3原色と色の3原色

光源から出た光を合成すると明るくなり白に近付いていきます。これに対して絵の具やインクをどんどん重ね合わせていくと光を吸収し黒に近付いていきます。その理由は、絵の具やインクによって見える色は、反射光を見ているからです。つまり青く見える絵の具は、青以外の色を吸収しているということです。

光のように、合成していくと明るくなるものを加算混合といいます。これに対し絵の具のように重ねるとどんどん暗くなるものを減算混合といいます。

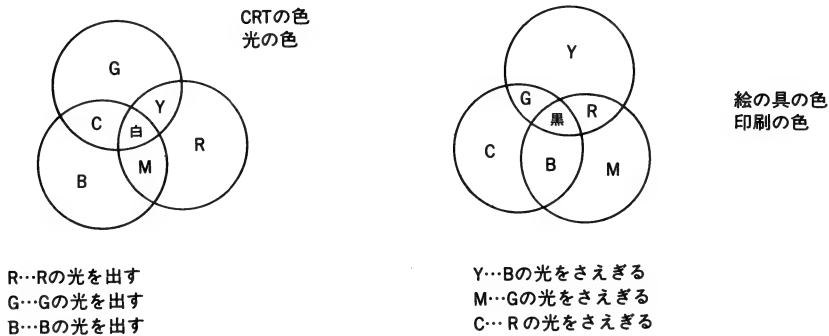


図1-26 RGBとYMCの違い

以下に加算混合，減算混合の例を示します。

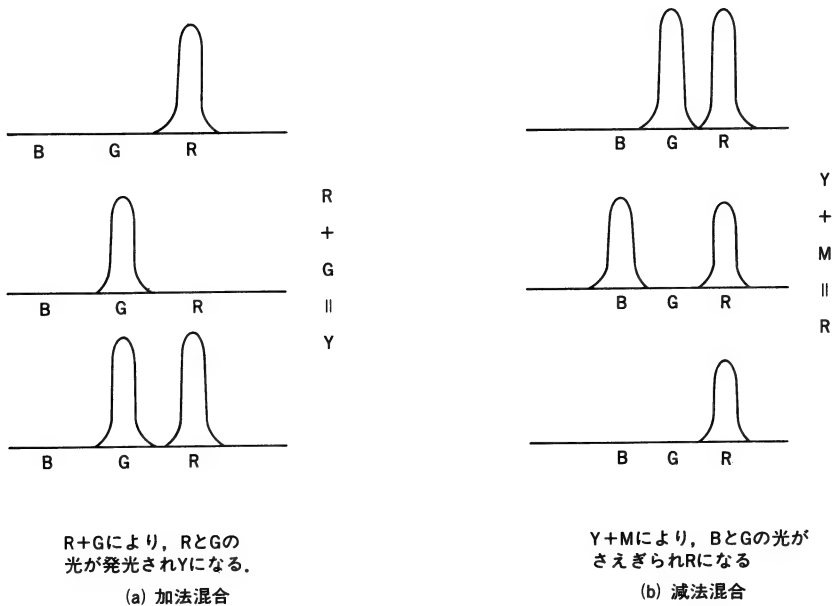


図1-27 加法混合と減法混合の違い

1-7 ディザリング

X1turboZ では、最高4096色の色しか表示できませんが、ディザリング機能を用いると疑似的に1670万色を同時に表示することができます。

普通、疑似的に色数を増やす方法として、タイリング方式とディザリング方式がありますが、タイリング方式は、1ドットの色の色忠実性は高いのですが、解像度が落ちてしまうため、あまり実用性はありません。

ディザリング機能は、あくまでも疑似的に色数を増やすため、中間色を使用すると幾何学模様が表示されることがあります。

ディザリング方式にもいろいろありますが、本節では 4×4 のディザマトリクスを使用する場合を例にとって説明します。

$$D = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}$$

ここでDをディザマトリクスといいます。

表示したい点の座標値を(x,y)，表示したい色をR,G,Bとします。ここでRGBは、画面座標系(x,y)の0-15で表される各色の輝度レベルを表します。

$$i = x \text{ AND } 3$$

$$j = y \text{ AND } 3$$

上式で定義されるi,jによりD(i,j)を求めます。

ディザリング後の色を(R',G',B')とすると、

$$R' = \frac{(R * 16 + D(i,j))}{16}$$

$$G' = \frac{(G * 16 + D(i,j))}{16}$$

$$B' = \frac{(B * 16 + D(i,j))}{16}$$

となります

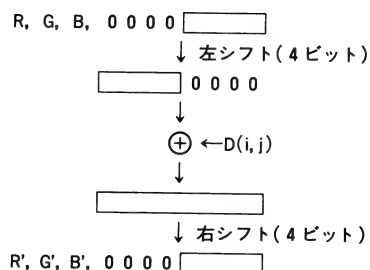


図1-28 ディザリングのしくみ

ディザ値によって輝度が高くなる確率が変わるため、色の細かい変化を表現することができます。

●ディザリングを使用したときの画面

VRAM が 2 枚のとき、カラーコード 0 を透明色として使用すると考えると、階調は、 $2^2 - 1 = 3$ 階調となります。つまり、輝度が 0%, 50%, 100% の 3 通りしかありません。次にこれを #0%, #50%, #100% として表してディザリングを行ったときと、行なわない時の違いを示します。

①ディザリングなしのとき

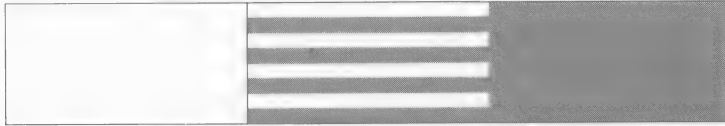


図1-29 ディザリングなしのときの表示

②ディザリングを使用したとき

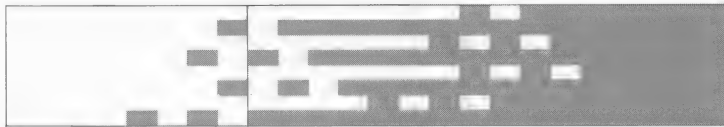


図1-30 ディザリングを使用したときの表示

遠くから見るとディザリングを使用したときは、除々に色が変化しているように見えます。

第2章

グラフィックツールの製作

本章では、コンピュータグラフィックスに必要な各種の描画アルゴリズムを解説するとともに、多色モードでのグラフィックサブルーチンを提供します。グラフィックサブルーチンは、アルゴリズムの理解を目的としているためすべて BASIC で記述しています。その結果、実行速度は決して速いとはいえませんが、IN,OUT 文を多用するなど機械語化しやすいコーディングになっています。従って、アルゴリズムを理解された読者は、機械語化を試みるのもよいでしょう。

2-1 各種描画アルゴリズム

2-1-1 直線描画のアルゴリズム

直線描画のアルゴリズムには、いろいろありますが、ここでは Bresenhan のアルゴリズムについて説明します。直線を引くアルゴリズムにおいては、いかにしてもっとも効率よくとぎれない線を引くかが重要になってきます。この Bresenhan のアルゴリズムでは、直線を描画する方法が、 x 方向の変化量と y 方向の変化量によって大きく分けられます。

これは、変化量の大きい方を基準にした方が、分割数が増え、きれいな線が、描けるからです。

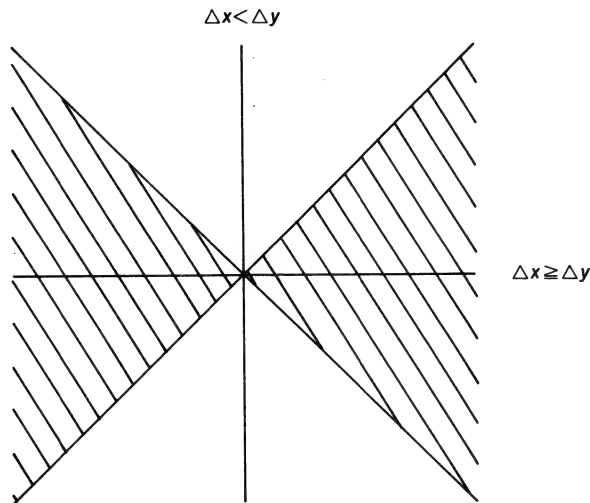


図2-2

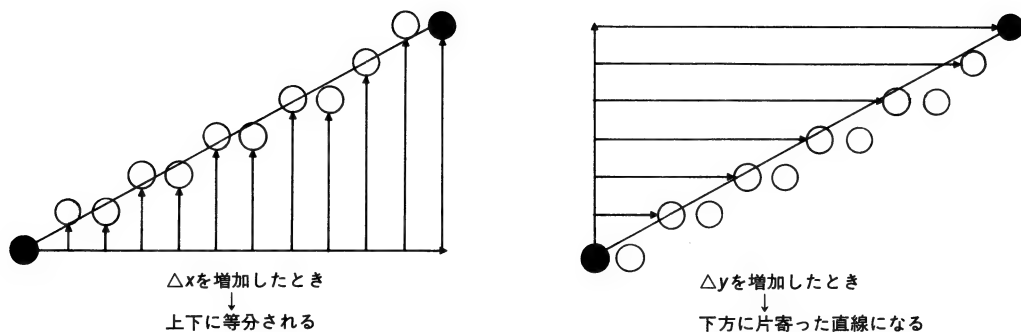


図2-3

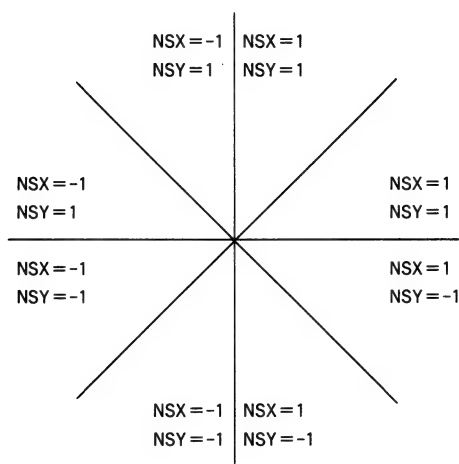


図2-4 xとyの増分値

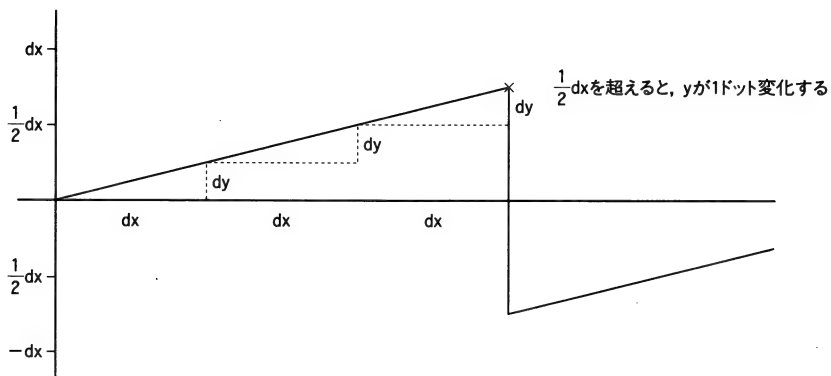


図2-5 誤差項εの変化

直線を描画しようとするとき、変化量の大きい方を1ドット増加または減少させた場合に、もう片方が1ドット増加または、減少するのか、変化しないのかを、誤差項NEで判定します。つまり、誤差項NEが、NDXより大きいか小さいかによって1ドット変化させるかどうかが決まるわけです。

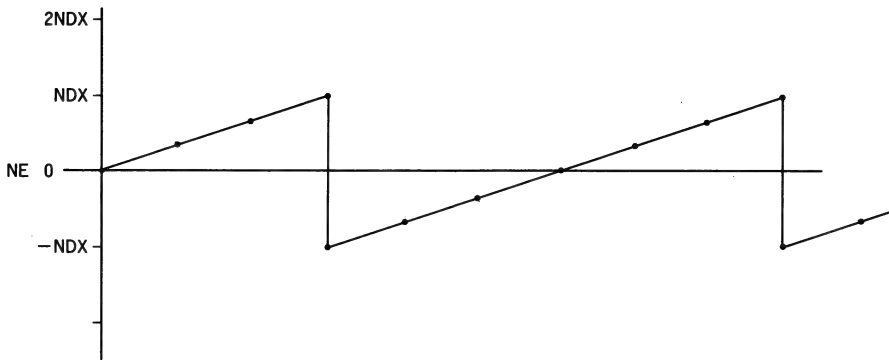


図2-6 整数型誤差項NEの変化

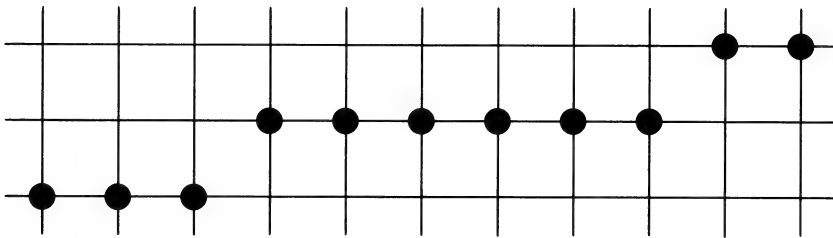


図2-7 画面上のドット表示

2-1-2 グラデーション付きの直線描画

直線を描画するとき、始点と終点の座標値と、直線の色を指定すると、その直線は同じ色で表示されます。ここでは、始点と終点に対して別の色を指定することにより、その間の色を補間し、グラデーションをつけて描画する方法について説明します。

直線を描画するとき、 x と y の変化量が大い方の値を1ドットずつ変化させた方が、きれいな線が描けることはすでに説明しましたが、色についても同じことが言えるため、同様の方法で補間します。本サブルーチンでは、色は、RGBで表すような仕様になっていますので、RGBそれぞれに対して、補間していきます。HSVやカラーコードで表現している場合も、同様にできます。ただし、HSVの場合、HUEは 359° の次が 0° になるため、注意が必要です。

2-1-3 グラデーション付きの3角形表示アルゴリズム

3角形の頂点の座標値と色を与えることにより、グラデーションを付けて表示する方法は、曲面の多面体表示など3次元グラフィックスではよく使われています。しかしイラストを描くときにも、3点の座標と、色を指定するだけで、その内部にグラデーションをつけて塗りつぶす機能があれば、非常に使いやすいものとなります。そこで、そのアルゴリズムについて説明します。このアルゴリズムは、グーローシェーディングと呼ばれる手法を参考にしています。

画面座標系での3点 $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$ とそれらの点の色 $C_1(R_1, G_1, B_1)$, $C_2(R_2, G_2, B_2)$, $C_3(R_3, G_3, B_3)$ が与えられているとします。ここで、3角形 $P_1P_2P_3$ の内部にある任意の点 $P(x, y)$ の色を求めてみます。まず、 y 座標について、ソートすると、次のようになります。

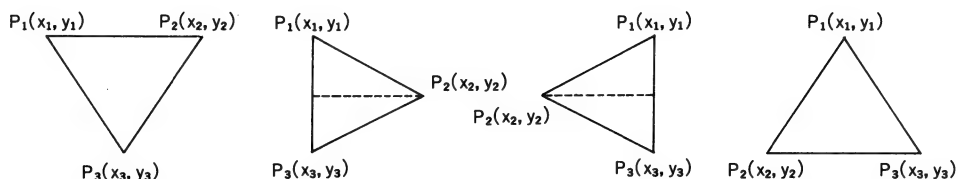


図2-8 三角形の種類

点 P_2 を通る水平線と辺 P_1P_3 との交点を $P_4(x_4, y_4)$ とすると3角形 $P_1P_4P_2$ と3角形 $P_3P_2P_4$ に分けられます。

点 P を通る水平線と3角形との交点を $P_5(x_5, y_5)$, $P_6(x_6, y_6)$ とします。点 P_5 の色と x 座標は、 P_1P_3 を線形補間することにより求めることができます。点 P_6 の色は、 $y_6 \leq y_2$ のときは P_1P_2 を線形補間し、 $y_6 > y_2$ のときは、 P_2P_3 を線形補間することによって求めることができます。点 P_5P_6 の色が決まれば、グラデーション付のラインと同じ方法で水平線を描画していけばよいでしょう。

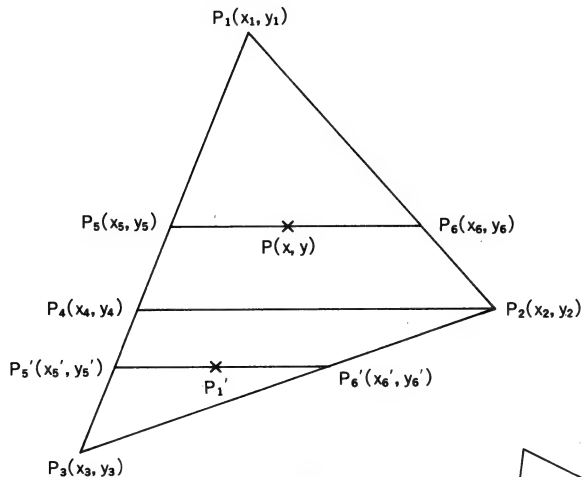


図2-9 描画方法

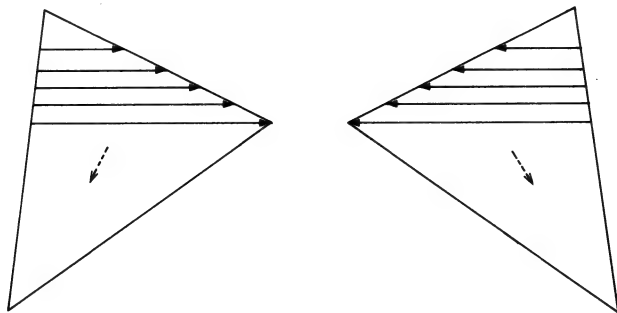


図2-10 描画順序

2-2 グラフィックサブルーチン

2-2-1 本グラフィックサブルーチンの特徴

- MERGE コマンドによるモジュール結合
- GOTO 文と REM 文によるブロック IF 文
- 変数名の命令規則によるモジュール変数の分離
- 他言語への 1 対 1 対応の移植性
- コメント行によるドキュメント化
- ラベル付きサブルーチンによる機能モジュール提供
- DEFINT, etc による, 変数の型宣言

(1) MERGE コマンドによるモジュール結合

本グラフィックサブルーチンは行番号が10000行から始まっており、ユーザーの作成するメインルーチンの行番号と重複しないようになっています。ですからメインルーチンは、1～9999で作成し、必要なモジュールを MERGE すればプログラムが完成するようになっています。

MERGE したままセーブすれば、次回からはそのまま実行できます。

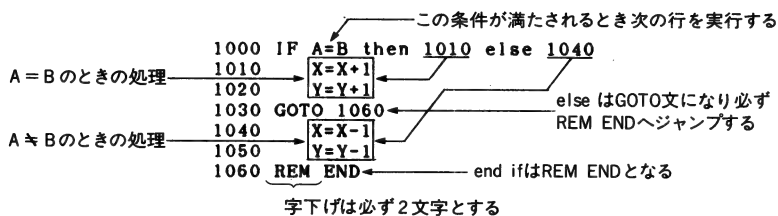
変数名規則さえ守っていれば MERGE したサブルーチンのモジュール変数は、全く意識しなくて構いません。

(2) GOTO 文と REM 文によるブロック IF 文

BASIC では、IF 文を複数行に渡って記述することができません。そのために、プログラムが分かりにくくなったり、他言語への書き換えが難しくなったりしています。そこで GOTO 文と REM 文を用いた、ブロック IF 文により記述します。

ブロック IF に文は次の条件があります。

① フォーマットは次の通りです



C 言語での記述

```

if (a==b) {
    x=x+1;
    y=y+1;
}
else {
    x=x-1;
    y=y-1;
}

```

FORTRANでの記述

```

IF (A.EQ.B) THEN
    X=X+1
    Y=Y+1
ELSE
    X=X-1
    Y=Y-1
ENDIF

```

② 原則としてブロック IF 文の長さは、画面の 1 ページ(25行)以内とします。画面上でのデバッグ効率と、モジュール構造を分かりやすくするため、1 ページ以内で理解できるようにするわけです。1 ページを超えるときは、なるべく、サブルーチン化するようにします。

これにより GOTO 文のジャンプ先も 1 画面で見られるため、デバッグしやすくなります。

③ GOTO 文は、コメント行へジャンプするようにします。

コメント行以外は、追加削除など、修正が加えられる可能性が大きいため GOTO 文はコメント行へジャンプするようにします。コメント行を記述したくないときは行番号の追加、削除のときに要注意です。

(3)変数名の命名規則

本サブルーチンパッケージでは、モジュール変数という概念を用いています。FORTRAN, C 言語などでは、グローバル変数とローカル変数という概念があり、プログラム作成時に、ローカル変数として、他のモジュールの変数名を気にせずプログラムを作ることができます。しかし、BASIC では、グローバル変数しかなく、サブルーチンで、どのような変数を使用するのかを常に認識していなければなりません。そこで、モジュール変数という概念を用いて、他モジュールの変数を意識せずにプログラムを作成する方法を説明します。

● BASIC でエラーとなる例

@リスト 2-2 入る

```
FOR I=1 TO 10
  GOSUB "GLINE"
NEXT
END
;
LABEL "GLINE"
FOR I=1 TO IX
  PSET(IX,IY)
NEXT
RETURN
```

同じ変数を使用すると
正常に動作しない!

①変数名の命名規則

BASIC 独特の%, \$, #などによる変数宣言は行わず、変数の開始文字でデータ型を区別します。また、一般変数、モジュール変数により、開始文字を区別します。

	一般変数	モジュール変数
整数型	I (Integer)	N (iNteger)
実数型	R (Real)	E (rEal)
倍精度型	D (Double)	A (dAburu)
文字型	C (Character)	H (cHaracter)

※doubleの0は0と誤り易いため
ローマ字記述のAを用います

1 文字目をとる 2 文字目をとる

表 2-1

また、FORTRAN などへの移植を考えるのなら、変数名は 6 文字以内にする必要があります。本サブルーチンパッケージは、すべて 6 文字以内となっています。

さらに、BASIC では A \$ と A % と A (1) は別変数として扱えますが、他言語ではこのような記述はできないため、使わないようにします。

また、C 言語などへの移植を考えるのなら、配列は 0 からとっておいたほうが良いでしょう。つまり、すべてのメインプログラムの初めに

```
DEFINT I-N
DEFSNG R,E
DEFDBL D,A
DEFSTR C,H
OPTION BASE 0
```

と記述しておくわけです。

そして、プログラムを作成するときは、I,R,D,C で始まるようにすればよいわけです。

②ラベル名の命令規則

本サブルーチンパッケージでは、ラベル名は Graphic の G をとって、必ず G で始まるようになっています。従って、メインルーチンでは G で始まるラベルはなるべく使わないようにした方が良いでしょう。

(4)ラベル付きサブルーチンの使用法

実際に使用することを考えると、BASIC のサブルーチンではなく、CALL 文によるアセンブラプログラムの呼び出しにした方が高速で使い易いのですが、今回は、描画アルゴリズムの理解が、目的であるためサブルーチン形式としました。しかし、サブルーチンでは引数渡しができないため、命令規則に合ったモジュール変数に値を代入してから、サブルーチンを呼び出します。

このように記述しておけば、アセンブラ版の CALL 命令ができれば、すぐに置き換えが可能です。また、引数渡しが可能な、他言語への移植も容易です。

点を描画するとき

BASIC で

```
PSET( x,y,c )
```

と記述するものを

本サブルーチンパッケージでは

```
NX=x:NY=y:NC=c
```

```
GOSUB "GPSET"
```

という形態で表現します。

なお、FORTRAN や BASIC の CALL 文では

```
CALL GPSET( x,y,c )
```


C言語では

```
gpset( x,y,c )
```

と記述します。

2-2-2 汎用グラフィックルーチンと高速グラフィックルーチン

よく高速グラフィックパッケージという言葉を見かけますが、はたして、高速グラフィックパッケージは良いのでしょうか。もし、全く同じ機能を持つのであれば高速の方が良いことはいうまでもありません。しかし、高速であるということとは、なんらかの機能が欠けていると考えた方が良いでしょう。

マウスで、画面上の座標値を指示して、直線や、曲線を描くときは、ユーザー座標系などは必要ないため、ウィンドウ、ビューポート変換やクリッピングのロジックは全く必要ありません。しかし、CADなどで実数の座標値のモデルを表示するときには、拡大、縮小がスムーズにできなければいけないため、ユーザー座標系は必須の機能となっています。

このように用途に応じて必要な機能だけを取り出すことにより、高速化したものを、高速グラフィックルーチンと考えた方が良いでしょう。

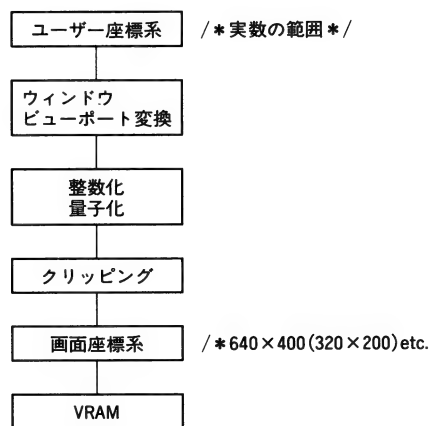


図2-1 汎用グラフィックパッケージの処理

2-2-3 グラフィックスモードの初期化処理

(1) 解像度の設定

画面の解像度は、WIDTH 文で設定します。

320×200 WIDTH 40,25,0,1

640×200 WIDTH 80,25,0,1

(2) V-RAM モードの設定

パレットモードか多色モードかの設定は、&H1FB0 で設定します。

多色モード(4096色×1面) OUT &H1FB0,&H80

多色モード(64色×2面) OUT &H1FB0,&H90

(3)スクリーンモードの設定

スクリーンについての初期設定をします。

4096色／64色(バンク0) OPTION SCREEN 0
SCREEN 0, 0

64色(バンク1) OPTION SCREEN 0
SCREEN 2, 2

2-2-4 パレットの初期化

カラーパレットの初期化コマンドは、用意されていないため、プログラムによる初期化が必要です。このパレットはリセットしても初期化されないの、注意しなければなりません。

電源 ON の時は、4096色モードとなっているため64色2画面モードにするときは、プログラムで初期化をするしかありません。そこで、64色2画面モード時のパレット初期化ルーチンと4096色同時表示時のパレット初期化ルーチンのプログラムを掲載します。

プログラムの説明をしますと、OUT &H1FB0,&H80 で、多色モードに設定していますが、これは多色モードでないと、パレットの初期化ができないためです。パレットの設定を行うときは、OUT &H1FC5,&H80 を実行してから設定しなければなりません。また、パレットの設定が終了したら、OUT &H1FC5,&H0 を実行しなければなりません。4096色の時はB,R,Gの順番にカラーコードと輝度を1つずつ上げていき、 $16 \times 16 \times 16 = 4096$ 色分のカラーコードを設定します。

それに対して、64色2面モードのときは、 $4 \times 4 \times 4 = 64$ を2回実行すればよいので、128色分のカラーコードを設定するだけでよいのです。

64色2面モードのときに、4096色分のカラーコードを初期化しないのは、おかしいと考えがちですが、64色2面モードのときは、片方のプレーンをマスクして2ビットしか見にいけないため、128色分の初期化で十分なのです。

2-2-5 BASIC で 4096 色同時表示を行う方法

XlturboZ は、4096色同時表示モードをサポートしていますが、BASIC の LINE コマンドや PSET コマンドで、カラーコード4095などと記述すると、Illegal function call というメッセージが出力されてしまいます。Z's STAFF で作成した画面を BASIC 上で表示することはできません、その上に絵を描くことはできません。そこで BASIC で4096色同時表示を可能にするサブルーチンを作成しました。

サブルーチンには、BASIC のグラフィックコマンドを使用した例と、VRAM を直接操作した

例の2つのサブルーチンがあります。前者は比較的高速(といっても一般のBASICより4倍遅い)で、ステップ数も短く理解しやすいようになっています。後者は、非常に遅く、ステップ数も多いのですが、VRAMを直接操作したり、アセンブラ命令と対応が取れるようになっているため、アセンブラ化すれば非常に高速になります。また、BASICの内部処理も理解できるようになっています。(但し、X1turboのBASICが、これと同じアルゴリズムであるかどうかは、確認していません)

このアルゴリズムでは、まず、SCREEN文で、SCREEN 3から描画していきます。SCREEN 3は、4096色モードの場合、最も、輝度の低いビットが、割り当てられています。そこで、NCで与えられたカラーコードの1ビット目を取り出します。取り出したビットが、BLUEならそのまま、REDなら2倍、GREENなら4倍すると、パレットモードのカラーコード(NCOL)になります。

	G	R	B
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

図2-11 パレットモードのカラーコード(初期値)

SCREEN 2は4096色モードの場合、2番目に輝度の低いビットが割り当てられています。そこで $(NC \text{ AND } 2) / 2$ の演算を実行することにより、2ビット目を取り出します。パレットモードへのカラーコードの変換は、SCREEN 3と同じです。

SCREEN 1,SCREEN 0についても、SCREEN 2の2を4、8に変更するだけで全く同じように変換できます。

このようにして、長方形や、円や、円弧なども、4096色モードで描画することができます。当然ビデオ入力した画像の上に重ねて描画することもできるようになります。

しかし、LINEを1本描画するために、このようなコーディングをするのは大変なので、サブルーチンとしておき、座標値と色コードだけ設定するようにすれば使いやすでしょう。

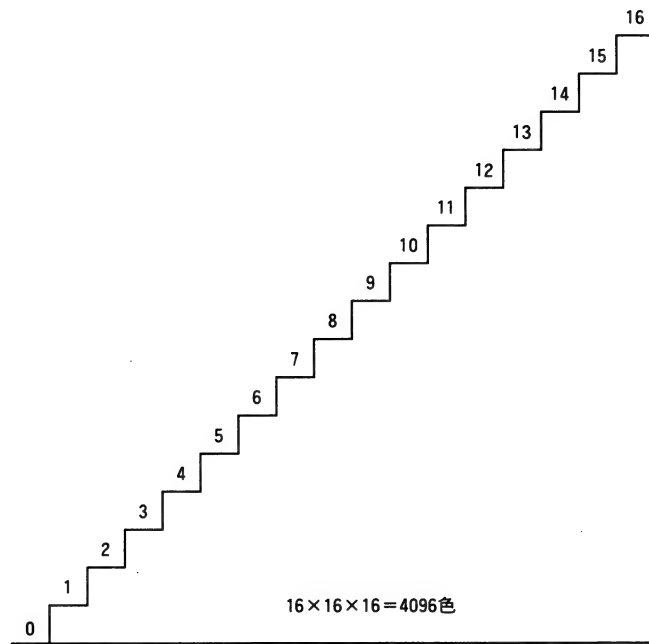


図2-12 4096色の輝度レベル

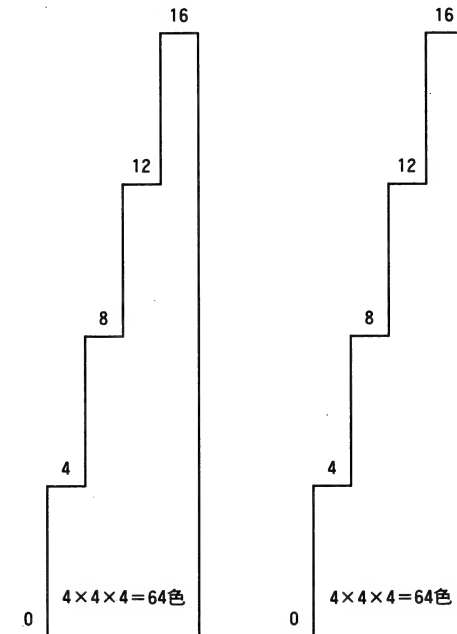


図2-13 64色2面モードのときの輝度レベル

第3章

ステレオグラフィックスの理論

3-1 3次元表示の方法

コンピュータを利用して3次元表示を行う場合、実際に利用されている方法として、現在以下の4通りがあります。

- ①アナグリフ方式(赤青スクリーン方式)
- ②ステレオスコープ方式
- ③偏光板による方式
- ④液晶シャッタによる時分割方式

この他に、ホログラフィなどの方法もありますが、ラスタスキャン型ディスプレイでの実現が難しいためここでは、述べないことにします。

これらの方法はいずれも、左目用の画像と右目用の画像を作成し、それを各種のフィルタを使用して分離するもので、両眼視差方式と呼ばれます。従って、どの方式も、特殊なめがねを必要とします。

めがね不要の方式として、レンチキュラ板や、IPを使用する方法がありますが、いずれも電氣的な信号処理が必要のため除いてあります。

ここでそれぞれの方式の利点、欠点をまとめると次表のようになります。

	めがね	カラー	視 点	観 察 者	画面の大きさ	ハードウェア
アナグリフ方式	必要 赤青スクリーン	不可	自由 (ただし映像は変化しない)	複数	1/1	なし
ステレオスコープ方式	必要 レンズ付 (慣れればめがね不要)	可	固定	1人	1/2	なし
偏光板方式	必要 偏光レンズ	可	自由 (ただし映像は変化しない)	複数	1/1	ディスプレイに偏光板とスクリーンが必要
液晶シャッタ方式	必要 液晶シャッタ	可	自由 (ただし映像は変化しない)	複数	1/1	簡単な装置が必要
レンチキュラ板IP方式	不要	可	自由 (視点を変えると映像を変化する)	複数	1/1	複雑な装置が必要

表3-1 3次元立体映像の分類

3-1-1 アナグリフ方式

この方式は一般に、赤青スクリーン方式として広く知られています。この方式では、左目用の映像を赤、右目用の映像を青で、画面上に重ねて表示します。赤いスクリーンは赤い光しか通さず、青いスクリーンは青い光しか通さないため、左右の目で別々の映像を見ることができます。

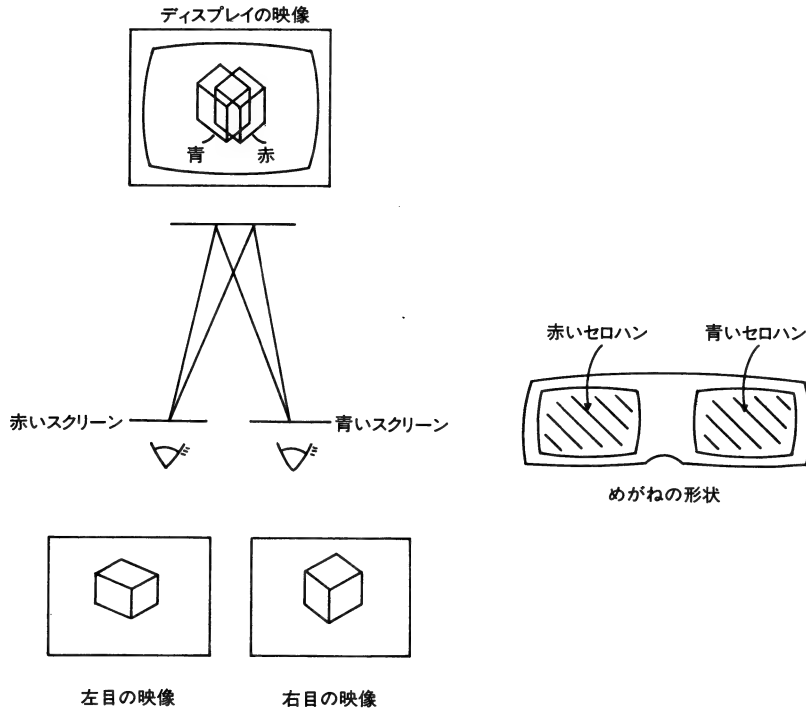


図3-1 アナグリフ方式

アナグリフ方式の利点は、赤と青のセロハンを使えばよいので、容易にめがねが作れることにあります。また、赤と青の2色しか使わないため、カラー信号の処理がやり易く、コンピュータディスプレイ上での実現も容易で、様々な映像を作ることが可能です。

しかし、ただ赤や青といっても光のスペクトルが少しずつ異なっているため、映像が2重になるゴーストが発生する可能性があります。また、色フィルタを使っているためカラー表示はできません。

さらに、赤と青の画像を長時間見ていると目が疲れるという欠点もあります。

3-1-2 ステレオスコープ方式

ステレオスコープ方式では、ディスプレイの画面をたてに2分割し、左側に左目用の画像、右側に右目用の画像を表示します。これを、画面の中心位置から見ることによって、立体的に見ることができます。

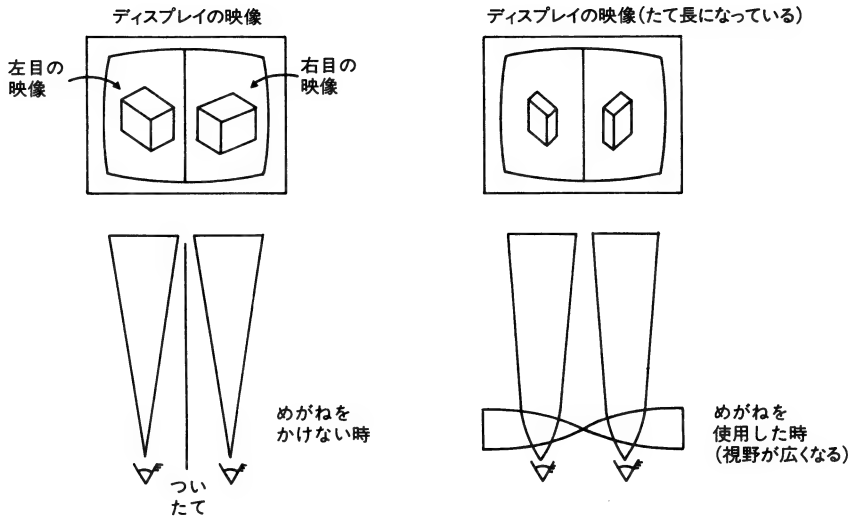


図3-2 ステレオスコープ方式

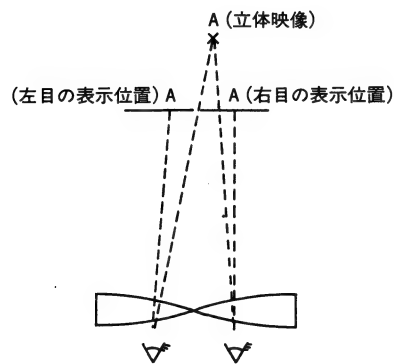


図3-3 立体映像の原理

ステレオスコープ方式の場合、画面を2分割するためそのままでは画面が小さくなってしまいますが、レンズ付きのメガネを使用すれば解決できます。

ステレオスコープ方式は、カラーで3次元表示を行うための最も簡単な方法として知られています。

ただし、この方式の場合、視点が、画面の中央に固定されてしまうため、一人しか見るできません。

3-1-3 偏光板方式

偏光板方式とは、スクリーンに偏光方向の異なる左右の画像を表示し、観察者が偏光レンズ付きのメガネをかけることによりカラーの立体映像を見ることができる方式です。パソコンなどのラスタスキャン型のディスプレイで、この方式を実現するには、TV画面に偏光をコントロールできる大型液晶シャッターを取り付けなければならない、現在ではまだまだ高価です。

3次元立体映画では、スクリーンに偏光の異なる映像を撮影することにより、かなり実用化されています。

偏光には、直線偏光と、円偏光がありますが、直線偏光は傾くと、画面が見えなくなる欠点があり、円偏光方式の方が優れています。

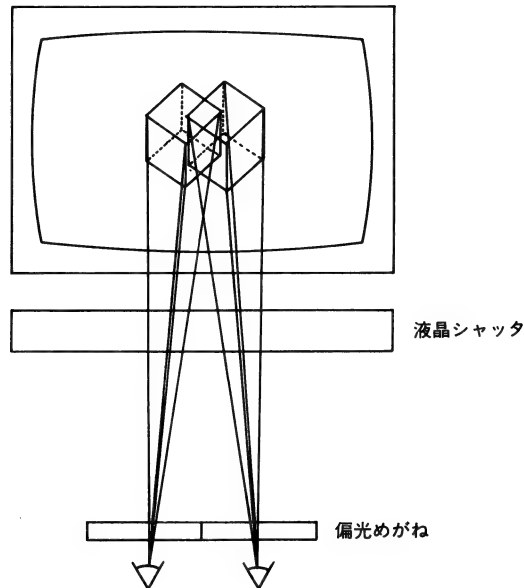


図3-4

3-1-4 時分割方式

時分割方式は、左目と右目の映像を交互に表示して液晶シャッター付きめがねを、画面と同期をとりながら ON/OFF にする方式です。VHD ビデオディスク、X1turbo など採用している方式で、価格も安く、現在最も普及しています。

しかし、時分割方式のため NTSC 信号では画面がちらつき、あまり画質は良くありません。また、テレビ信号と同期をとって液晶シャッターをコントロールする必要があるため、めがねについているコードがわずらわしく感じます。

画質の悪さについては、EDTV が実現すればノンインターレース方式で表示できるため、かなり明るくなり、ちらつきも減っていくと思われます。そして、液晶シャッターの開閉速度が向上すれば、もっとちらつきのない画面が、表示できるようになるでしょう。

3-2 3次元立体表示の仕組み

人間はどのようにして、遠近感を得ているのでしょうか。

人間は、さまざまな状況のもとで3次元的な距離を認識します。その中の主なものとして以下に示すものが、挙げられます。

- ①レンズの焦点距離によるもの
- ②両眼視差によるもの
- ③物体の大きさなどによるもの
- ④物体速度によるもの

①レンズの焦点によるもの

人間がものを見るとき、その物体に焦点を合わせようとしますが、そのときのレンズの膨らみ具合により距離を測るものです。ですから片目であっても距離は分かりますが、遠すぎるものに関しては分かりません。

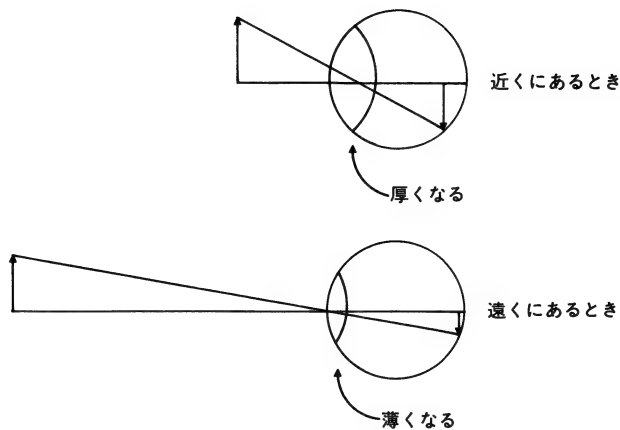


図3-5

②両眼視差によるもの

遠くにあるものと近くにあるものでは、左右の目に見える見え方が異なります。つまり、遠くになればなるほど、左目と右目の映像は同じになります。3次元立体表示機能は、ほとんどこの両眼視差を利用した方式です。

人間の目は、左右の画像のパターン認識を行い、パターンの一致した部分の左右の間隔により、距離を測ります。

X1 turboなどの液晶シャッタによる方式は、この両眼視差によるものです。

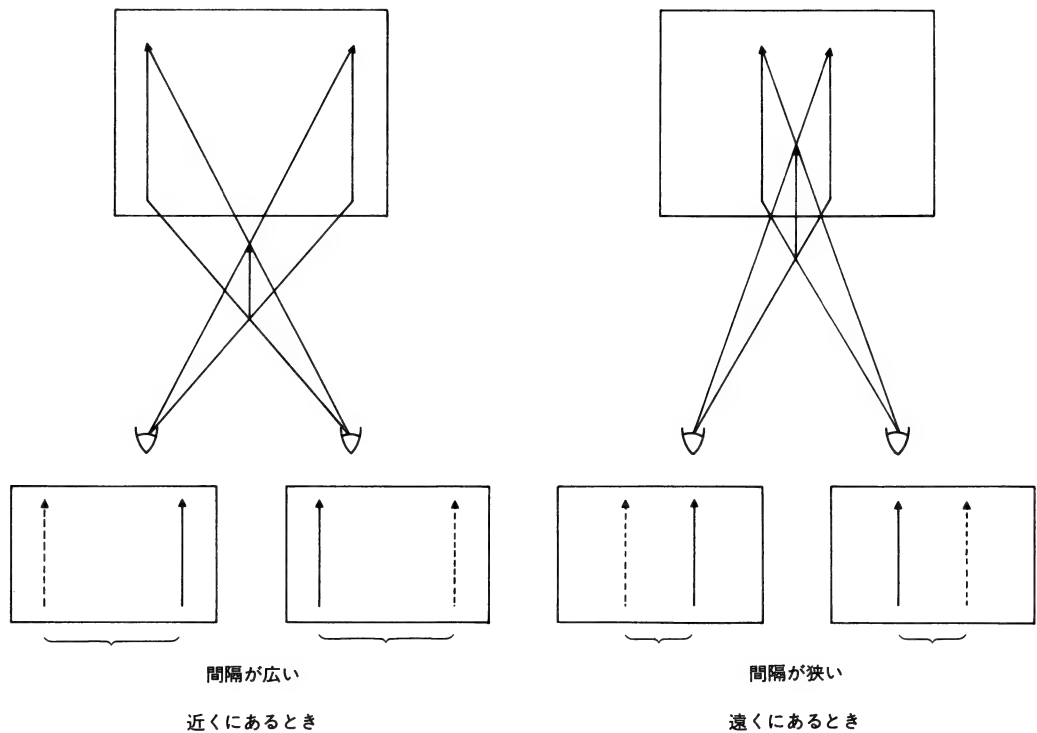


図3-6

③物体の大きさによるもの

同じ大きさのものであれば、手前のものほど大きく見え、遠くにあるものほど小さく見えます。
絵画が、立体的に見えるのは、この大きさによる場合が多いです。

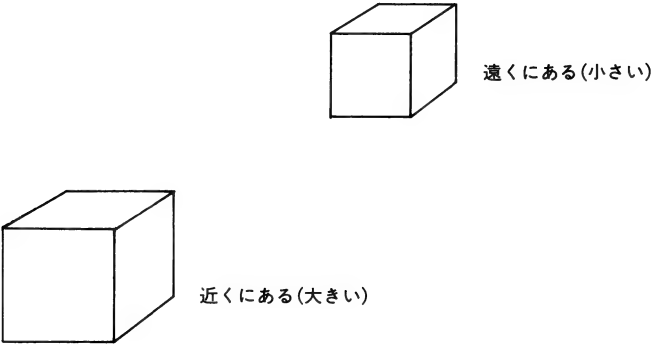


図3-7

④移動速度によるもの

手前のものほど早く動き、遠くのものほど遅く動くため、物体の移動速度でも、距離を知ることができます。

アニメーションなどでは、セル画と背景の移動速度を変えることにより、立体感を出そうとしています。

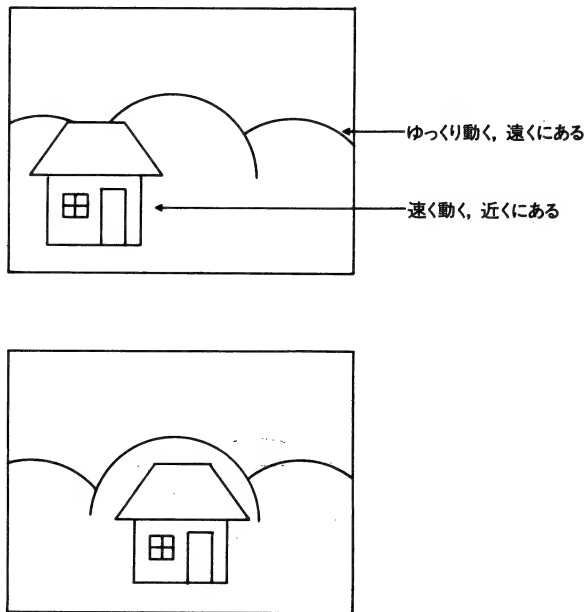


図3-8

第4章

FM音源によるMUSIC機能

4-1 光と音

光も音も波の一種です。光は、可視光線ともいわれ、周波数が非常に高い波です。それに対し、音は音波ともいわれ、周波数が低い波です。つまり、音も光も周波数が異なるだけで、同じようなものなのです。

光も音もその周波数成分によって、その性質が異なります。光は、主に色でその性質を表しますが、音は、音色(音の種類)、音程(音の高さ)、音量(音の大きさ)によって、その性質を定義します。これらを音の3要素といいます。

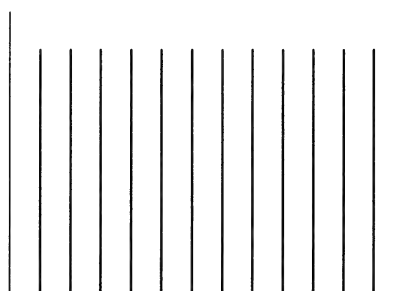
4-2 音色とは

音色とは、音の種類のことです。例えば、ピアノの音とかバイオリンの音とかトランペットの音などと呼ばれているのは、音色のことです。色を光のスペクトルで表したように、音色もスペクトルで表すことができます。スペクトルとは、音の周波数成分の分布のことです。つまり、どのような周波数の音がどれだけ含まれているかを表したものです。正弦波は、一つの周波数成分しかありませんが、楽器などの音は、基準となる音の2倍、3倍・・・の周波数の音が、含まれています。これらの音を2倍音、3倍音・・・と呼び、基準となる音を基音と呼んでいます。音色は、これらの倍音成分の含まれている割合により変わるわけです。

普通の楽器は、整数倍の倍音からできていますが、打楽器などは、非整数倍の倍音も含まれます。非整数倍の倍音が増加すると、ノイズに近くなります。ノイズとは、すべての周波数成分を同じ様に持っている音のことです。



正弦波のスペクトル



ノイズのスペクトル

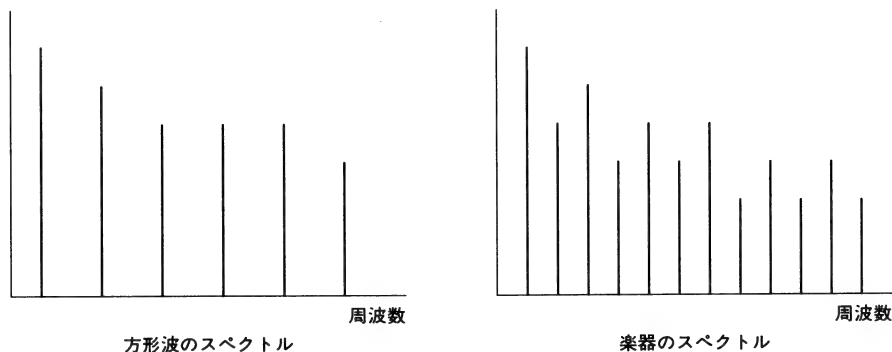
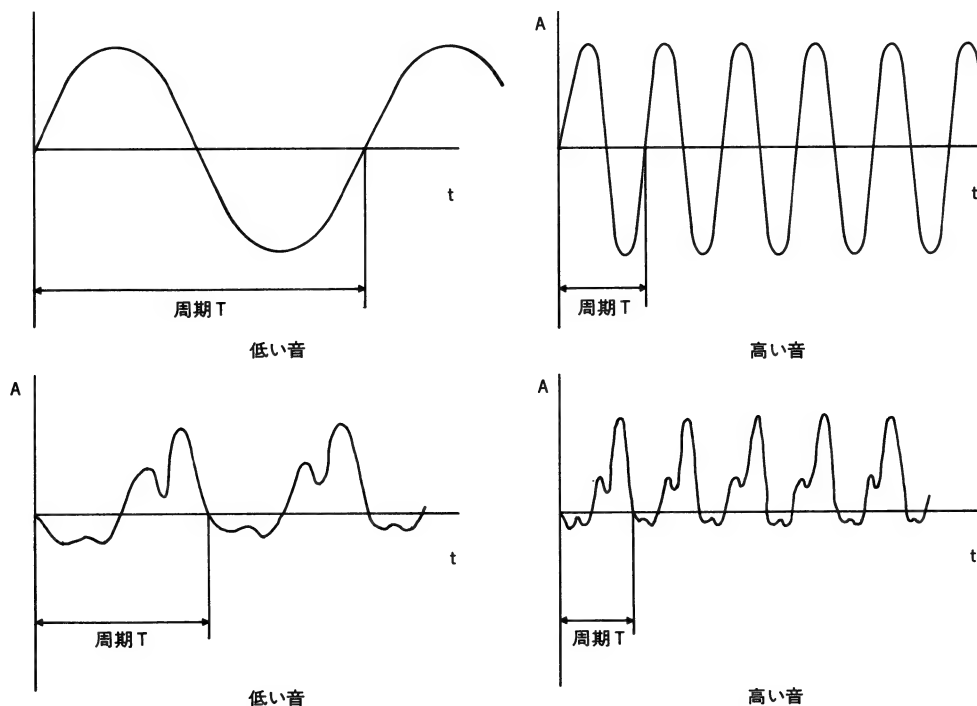


図4-1 音色とスペクトル

4-3 音程とは

音程とは、音の高さのことです。例えば、ド、レ、ミ、ファ、ソ、ラ、シ、ド、といえは音程を表します。音程は、音の周波数によって表されます。高い音ほど周波数が高く、低い音ほど周波数が低くなります。周波数はA3の音といえは、440Hz というように決められています。つまり、音色によって決められた周波数成分によってできる波形を、音程で指定された周波数で繰り返すわけです。



※周波数は $1/T$ で表わされます。

図4-2 音程の違いによる時間と振幅の変化

4-4 音量とは

音量とは、音の大きさのことです。音量は、音の振幅の大きさと表され、単位はdB または、ホンで表します。大きな音ほど振幅は大きく、小さな音ほど振幅は小さくなります。

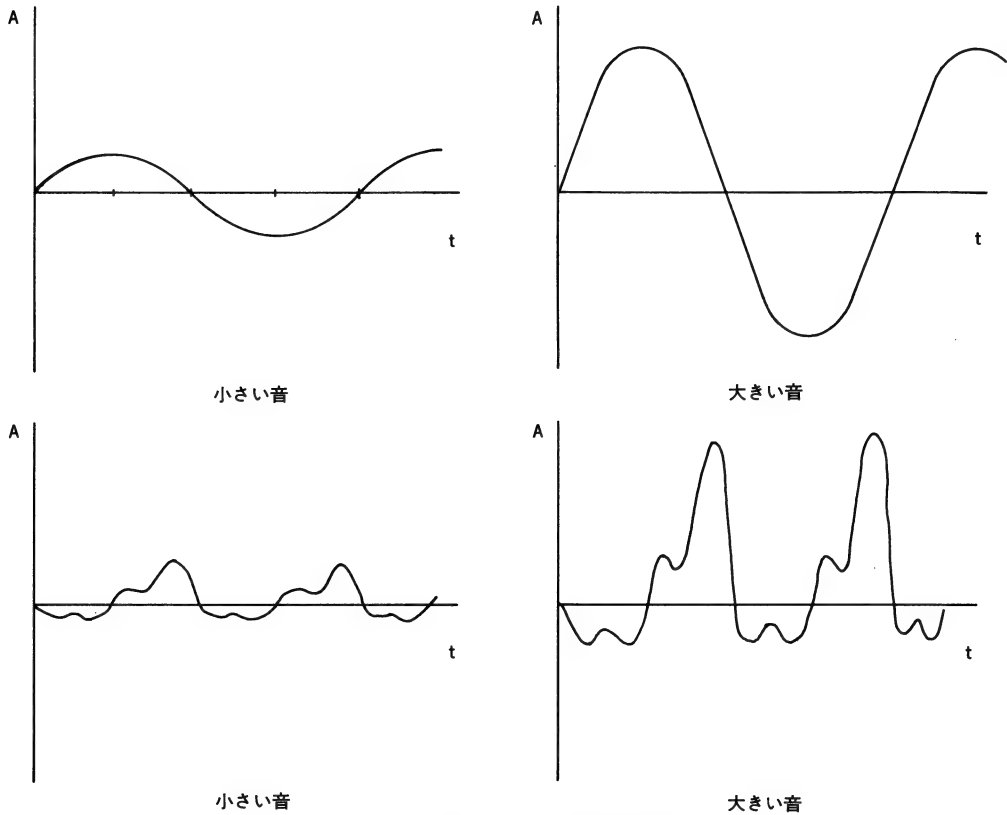


図4-3 音量の違いによる時間と振幅の変化

4-5 音色エディタの機能

音色エディタは、音を作り出すソフトです。ミュージックツールには、200種類もの音色が用意されているため、それだけでも十分いろいろな曲を作ることができます。

音は、適当に操作しても、全く出ないということはありません。しかし、適当に操作して、気に入った音ができたとしても、それは、音を作ったことにはなりません。音を作るとは、頭の中でどのような曲を作るのかを考えてからその音を作りだしていくものです。そのためにはFM音源のしくみについての知識が必要です。そこでFM音源について説明します。

4-5-1 オペレータとは

X1turboZには、オペレータが4つあります。この4つのオペレータを組み合わせることにより、いろいろな音色を作り出すことができます。オペレータにはキャリアとモジュレータがあります。しかし、オペレータとは、単なる正弦波を生成する発振器ですので、キャリア用のオペレータとか、モジュレータというふうに分かれてはいません。実際には、使用法によって、キャリアなのかモジュレータなのか決まります。つまり、キャリアは、音量を決定し、モジュレータは音色を決定します。

オペレータには、フィードバックできるオペレータが1つだけあります。フィードバックとは、オペレータが自分自身に対して変調をかけることです。これは、縦にオペレータを重ねたのと同じ意味になります。X1turboZでは、オペレータ1だけがフィードバックをかけることができます。変調とは、どのようなことなのかを理解するためにI部の最後に、キャリアとモジュレータの周波数により、音の波形がどのように変化するかを見ることのできるプログラムを載せてあります。

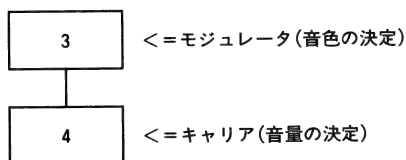
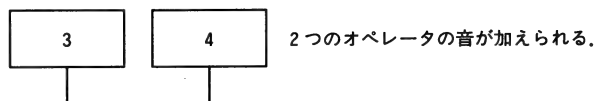


図4-4

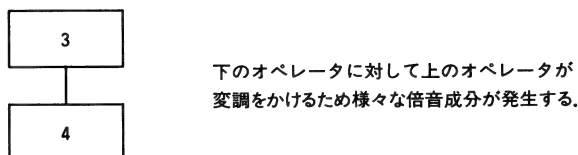
4-5-2 アルゴリズムとは

オペレータの組合せ方をアルゴリズムといいます。X1turboZでは、8つのアルゴリズムがあります。キャリアは、最下段に位置しています。また、モジュレータは二段目以上に位置しています。アルゴリズムには、いろいろな型があるので理解しにくいと思いますが、基本的には2つのオペレータの合成に置き換えて考えることができます。

* 並列に2つの音の合成



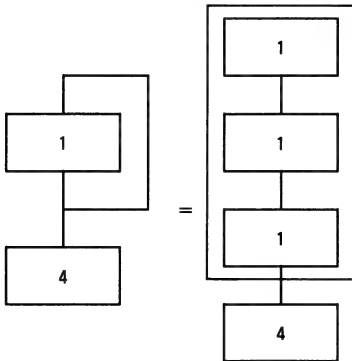
* 直列に2つの音の合成



このように、オペレータを並列に並べて合成すると2つのキャリアの周波数成分が加えられるだけです。それに対して、変調をかけたときは、様々な倍音声分を作り出すことができます。また、モジュレータに対して変調をかけることもできるため、変調効果を大きくしたいときは、オペレータを縦に直列に並べた方が良いです。

次の図に示すように、複雑なアルゴリズムも、2つのオペレータの合成に置き換えて考えることができます。

- フィードバックがあるとき



- 複数のモジュレータがあるとき

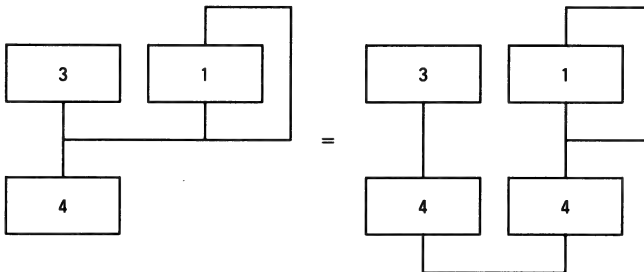


図4-7

- 複数のキャリアにフィードバックがあるとき

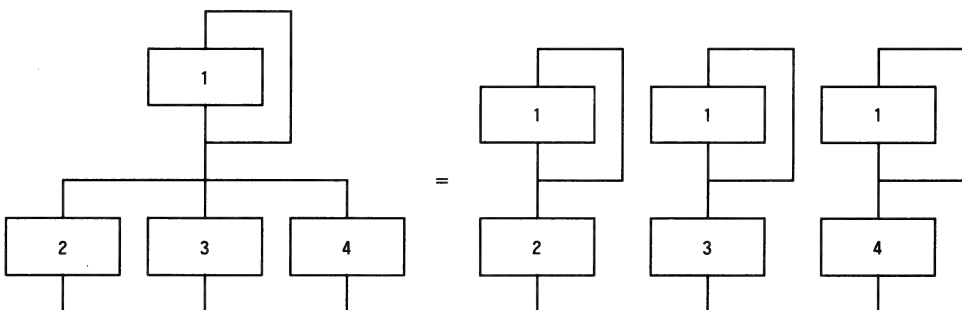


図4-8

4-5-3 エンベロープとは

音には、時間的な変化があります。例えば、ピアノの鍵盤をたたくと、その瞬間には最も大きくなり、それからキーを押している間は、ゆっくりと小さくなっていきます。そして、キーを離してから、しばらくすると音は聞こえなくなります。このような音の時間的な変化をエンベロープといいます。楽器の音の場合、音程がずれては困るため、音程は一定ですが、音量と音色は、時間的に変化します。音量を変化させるときは、キャリアに対してエンベロープを効かせます。それに対して音色を変化させるときは、モジュレータに対してエンベロープを効かせます。

キャリアのエンベロープ : 音量の時間的変化

モジュレータのエンベロープ : 音色の時間的変化

エンベロープは、AR,1DL,1DR,2DR,RRの5つのパラメータによって設定されます。最後にRのつくものは、RATEといい、速度を表します。これに対して、Lのつくものは、LEVELといい大きさを表します。

また、音の鳴り始める時間、つまりピアノの鍵盤をたたいた時間や、トランペットを吹き始めた時間を、KEY ONといいます。これに対して、音の鳴りやむ時間、つまりピアノの鍵盤を離れた時間や、トランペットを吹くのをやめた時間を、KEY OFFといいます。

1. AR (ATTACK RATE)

ARとは、KEY ON から、音が最大になるまでの速度のことです。音の立ち上がりを決めるパラメータで、ピアノなどは大きく、バイオリンなどは小さくなっています。

2. 1DL (1st DECAY LEVEL)

音は、KEY ON の瞬間、大きな音がでますが、その後持続状態が保たれます。1DLとは、持続状態が始まるときの、音のレベルのことです。

3. 1DR (1st DECAY RATE)

1DRとは、最大レベルから、1DL で設定したレベルになるまでの速度のことです。

4. 2DR (2nd DECAY RATE)

2DRとは、持続状態での音の減衰する速度のことです。2DR があまりに大きすぎると音はよく出なくなってしまいます。

5. RR (RELEASE RATE)

RRとは、KEY OFF から音が聞こえなくなるまでの速度のことです。RR が小さい程、余韻が残ります。

グラフィックサブルーチン一覧

ウィンドウ・ビューポート変換のプログラム

```
10  name "WINDOW"
1000 '
1010 ' WINDOW VIEW PORT CONVERT PROGRAM
1020 '   Programmed by Joe Masumura
1030 '
1040 ' VARIABLE INITIALIZE
1050 '
1060 '   IVX1 : ビューポート ノ ヒタリウエ ノ X サ`ヒョウチ
1070 '   IVY1 : ビューポート ノ ヒタリウエ ノ Y サ`ヒョウチ
1080 '   IVX2 : ビューポート ノ ミキ`シタ ノ X サ`ヒョウチ
1090 '   IVY2 : ビューポート ノ ミキ`シタ ノ Y サ`ヒョウチ
1100 '   IWX1 : ウィントウ ノ ヒタリウエ ノ X サ`ヒョウチ
1110 '   IWY1 : ウィントウ ノ ヒタリウエ ノ Y サ`ヒョウチ
1120 '   IWX2 : ウィントウ ノ ミキ`シタ ノ X サ`ヒョウチ
1130 '   IWY2 : ウィントウ ノ ミキ`シタ ノ Y サ`ヒョウチ
1140 '   IX1  : チョクセン ノ シデン ノ X サ`ヒョウチ
1150 '   IY1  : チョクセン ノ シデン ノ Y サ`ヒョウチ
1160 '   IX2  : チョクセン ノ シュウデン ノ X サ`ヒョウチ
1170 '   IY2  : チョクセン ノ シュウデン ノ Y サ`ヒョウチ
1180 '
1190 DEFINT I-N ' 変数宣言
1200 DEFSNG R,E
1210 DEFDBL D,A
1220 DEFSTR C,H
1230 OPTION BASE 0
1240 '
1250 ' GRAPHIC INITIALIZE ' グラフィックスの初期化
1260 '
1270 OPTION SCREEN 0
1280 SCREEN 0,0
1290 WIDTH 80,25,1,0
1300 WINDOW
1310 KLIST 0
1320 CONSOLE 0,25
1330 '
1340 ' ビューポート ノ セツテイ
1350 '
1360 CLS
1370 PRINT "VIEW PORT (X1,Y1,X2,Y2)" ' ビューポートの入力
1380 INPUT IVX1,IVY1,IVX2,IVY2
1390 '
1400 IF IVX1>IVX2 THEN 1360 ' エラーチェック
1410 IF IVY1>IVY2 THEN 1360
1420 IF IVX1<0 OR IVY1>399 THEN 1360
1430 IF IVY1<0 OR IVY1>639 THEN 1360
1440 IF IVX2<0 OR IVY2>399 THEN 1360
1450 IF IVY2<0 OR IVY2>639 THEN 1360
1460 '
1470 LINE(IVX1,IVY1)-(IVX2,IVY2),PSET,1,B ' ビューポートの描画
1480 '
1490 ' ウィントウ ノ セツテイ
1500 '
1510 PRINT "WINDOW (X1,Y1,X2,Y2)" ' ウィンドウの入力
1520 INPUT IWX1,IWY1,IWX2,IWY2
1530 '
1540 IF IWX1>IWX2 THEN 1460 ' エラーチェック
1550 IF IWY1>IWY2 THEN 1460
1560 '
1570 ' チョクセン ノ ヒョウカ`
1580 '
1590 PRINT "X1,Y1,X2,Y2" ' 座標値の入力
1600 INPUT IX1,IY1,IX2,IY2
```

```

1610 '
1620 IF IX1<IX1 OR IX1>IX2 THEN 1580 ' エラーチェック
1630 IF IY1<IY1 OR IY1>IY2 THEN 1580
1640 IF IX2<IX1 OR IX2>IX2 THEN 1580
1650 IF IY2<IY1 OR IY2>IY2 THEN 1580
1660 '
1670 GOSUB "GTRAN" ' 座標変換
1680 '
1690 LINE(IX1,IY1)-(IX2,IY2),PSET,5 ' 直線の描画
1700 '
1710 END
1720 '
1730 ' ウィンドウ ヒューポート へんかん
1740 '
1750 ' IX1 : デン 1 ノ X サ`ヒョウチ
1760 ' IY1 : デン 1 ノ Y サ`ヒョウチ
1770 ' IX2 : デン 2 ノ X サ`ヒョウチ
1780 ' IY2 : デン 2 ノ Y サ`ヒョウチ
1790 '
1800 LABEL "GTRAN"
1810 IX1=IX1-(IX2+IX1)/2 ' 平行移動
1820 IX1=(IX2-IX1)/(IX2-IX1)*IX1 ' スケール
1830 IX1=IX1+(IX2+IX1)/2 ' 平行移動
1840 '
1850 IY1=IY1-(IY2+IY1)/2 ' 平行移動
1860 IY1=(IY2-IY1)/(IY2-IY1)*IY1 ' スケール
1870 IY1=IY1+(IY2+IY1)/2 ' 平行移動
1880 '
1890 IX2=IX2-(IX2+IX1)/2 ' 平行移動
1900 IX2=(IX2-IX1)/(IX2-IX1)*IX2 ' スケール
1910 IX2=IX2+(IX2+IX1)/2 ' 平行移動
1920 '
1930 IY2=IY2-(IY2+IY1)/2 ' 平行移動
1940 IY2=(IY2-IY1)/(IY2-IY1)*IY2 ' スケール
1950 IY2=IY2+(IY2+IY1)/2 ' 平行移動
1960 RETURN
1970 '

```

RGBからHSVへの変換プログラム

```

10 ' name "RGB2HSV"
1000 '
1010 ' R.G.B. H.S.V. CONVERT PROGRAM
1020 ' Programmed by Joe Masumura
1030 '
1040 ' VARIABLE INITIALIZE
1050 '
1060 ' IH : Hue シキソウ (イロアイ)
1070 ' IS : Saturation サイト` (アサ`ヤカサ)
1080 ' IV : Value メイト` (アカルサ)
1090 ' IRGB : R.G.B. ヒョウケ`ン ノ イロ ノ キト`
1100 ' IMAX : R.G.B. ノ サイト`イ キト` ノ イロ ノ INDEX
1110 ' IMIN : R.G.B. ノ サイト`イ キト` ノ イロ ノ INDEX
1120 ' IN : R.G.B. ヒョウケ`ン ノ カイショウ スウ
1130 '
1140 DEFINT I-N ' 変数宣言
1150 DEFSTR R,E
1160 DEFDBL D,A
1170 DEFSTR C,H
1180 OPTION BASE 0
1190 '
1200 DIM IRGB(2)
1210 IN=15 ' X1-turboZの場合15階調
1220 '
1230 ' GRAPHIC INITIALIZE
1240 '
1250 WIDTH 40,25,0,1
1260 CONSOLE 0,15
1270 KLIST 0

```

```

1280 OUT &H1FB0,&H80
1290 '
1300 ' R.G.B. データ ノ ニュウリョク
1310 '
1320 CLS
1330 PRINT "ENTER R,G,B ( 0 -";IN;" )"
1340 INPUT IRGB(0),IRGB(1),IRGB(2)
1350 '
1360 FOR I=0 TO 2 ' エラ- ショリ
1370 IF IRGB(I)<0 OR IRGB(I)>IN THEN 1320
1380 NEXT
1390 '
1400 ' サイタ`イチ サイショウチ ノ ケイサン
1410 '
1420 IMAX=0 ' サイタ`イチ ノ ショキカ
1430 IMIN=0 ' サイショウチ ノ ショキカ
1440 FOR I=1 TO 2
1450 IF IRGB(I)>IRGB(IMAX) THEN 1460 ELSE 1470
1460 IMAX=I
1470 REM END
1480 '
1490 IF IRGB(I)<IRGB(IMIN) THEN 1500 ELSE 1510
1500 IMIN=I
1510 REM END
1520 NEXT
1530 '
1540 ' メイト` (Value) ノ ケイサン
1550 '
1560 IV=IRGB(IMAX)/IN*100
1570 '
1580 ' サイト` (Saturation) ノ ケイサン
1590 '
1600 IF IRGB(IMAX)=0 THEN 1610 ELSE 1630
1610 IS=0
1620 GO TO 1640
1630 IS=(1-IRGB(IMIN)/IRGB(IMAX))*100
1640 REM END
1650 '
1660 ' シキソウ (Hue) ノ ケイサン
1670 '
1680 IH=120*(IMAX)
1690 IF IRGB(IMAX)=IRGB(IMIN) THEN 1700 ELSE 1720
1700 IH=0
1710 GO TO 1790
1720 IF ((IMAX-1)+3) MOD 3 = IMIN THEN 1730 ELSE 1760
1730 IVAL=IRGB(((IMAX+1)+3) MOD 3)
1740 IH=IH+60*(IVAL-IRGB(IMIN))/(IRGB(IMAX)-IRGB(IMIN))
1750 GO TO 1790
1760 IVAL=IRGB(((IMAX-1)+3) MOD 3)
1770 IH=IH-60*(IVAL-IRGB(IMIN))/(IRGB(IMAX)-IRGB(IMIN))
1780 REM END
1790 REM END
1800 IH=(IH+360) MOD 360
1810 '
1820 ' H.S.V. データ ノ ヒョウジ`
1830 '
1840 PRINT "R,G,B =";IRGB(0);",";IRGB(1);",";IRGB(2)
1850 PRINT "H,S,V =";IH;",";IS;",";IV
1860 '
1870 OUT &H1FC5,&H80
1880 OUT &H1088,&H80+IRGB(2)
1890 OUT &H1188,&H80+IRGB(0)
1900 OUT &H1288,&H80+IRGB(1)
1910 OUT &H1FC5,&H0
1920 '
1930 LINE (0,150)-(319,199),PSET,7,BF
1940 '
1950 END
1960 '

```

HSVからRGBへの変換プログラム

```

10 ' name "HSV2RGB"
1000 '
1010 ' H.S.V. R.G.B. CONVERT PROGRAM
1020 '   Programmed by Joe Masumura
1030 '
1040 ' VARIABLE INITIALIZE
1050 '
1060 '   IH  : Hue シキソウ (イロアイ)
1070 '   IS  : Saturation サイトン (アサキヤカサ)
1080 '   IV  : Value ムイトン (アカルサ)
1090 '   IRGB : R.G.B. ヒョウケン ノ イロ ノ キト
1100 '   IN  : R.G.B. ヒョウケン ノ カイチョウ スウ
1110 '   IMAX : R.G.B. ノ サイトン イ キト ノ イロ ノ INDEX
1120 '   IMIN : R.G.B. ノ サイトン キト ノ イロ ノ INDEX
1130 '
1140 DEFINT I-N ' 変数宣言
1150 DEFSNG R,E
1160 DEFDBL D,A
1170 DEFSTR C,H
1180 OPTION BASE 0
1190 '
1200 DIM IRGB(2)
1210 IN=15 ' X 1 - t u r b o Z の場合 15 階調
1220 '
1230 ' GRAPHIC INITIALIZE
1240 '
1250 WIDTH 40,25,0,1
1260 CONSOLE 0,15
1270 KLIST 0
1280 OUT &H1FB0,&H80
1290 LINE (0,150)-(319,199),PSET,7,BF
1300 '
1310 ' H.S.V. データ ノ ニュウリョク
1320 '
1330 CLS
1340 PRINT "ENTER H(0-360),S(0-100),V(0-100)"
1350 INPUT IH,IS,IV
1360 '
1370 IF IH<0 OR IH>360 THEN 1330
1380 IF IS<0 OR IS>100 THEN 1330
1390 IF IV<0 OR IV>100 THEN 1330
1400 '
1410 ' サイトン イ キト ノ イロ ノ ケイサン
1420 '
1430 IMAX=((IH+60)*120) MOD 3
1440 IRGB(IMAX)=IN
1450 '
1460 ' サイトン キト ノ イロ ノ ケイサン
1470 '
1480 IMIN=((IH+60+180)*120) MOD 3
1490 IRGB(IMIN)=IN*(100-IS)/100
1500 '
1510 ' ノコリ ノ イロ ノ ケイサン
1520 '
1530 IF (IMAX+1) MOD 3 = IMIN THEN 1540 ELSE 1570
1540   IVAL=(IMAX+3-1) MOD 3
1550   IRGB(IVAL)=IRGB(IMIN)+(IRGB(IMAX)-IRGB(IMIN))*(60-(IH MOD 60))/60
1560 GO TO 1590
1570   IVAL=(IMAX+3+1) MOD 3
1580   IRGB(IVAL)=IRGB(IMIN)+(IRGB(IMAX)-IRGB(IMIN))*(IH MOD 60)/60
1590 REM END
1600 '
1610 FOR I=0 TO 2
1620   IRGB(I)=IRGB(I)*IV/100
1630 NEXT
1640 '
1650 ' R.G.B. データ ノ ヒョウケン
1660 '
1670 PRINT "H,S,V =";IH;" ";IS;" ";IV
1680 PRINT "R,G,B =";IRGB(0);" ";IRGB(1);" ";IRGB(2)
1690 '

```

```

1700 OUT &H1FC5,&H80
1710 OUT &H1088,&H80+IRGB(2)
1720 OUT &H1188,&H80+IRGB(0)
1730 OUT &H1288,&H80+IRGB(1)
1740 OUT &H1FC5,&H0
1750 '
1760 LINE (0,150)-(319,199),PSET,7,BF
1770 '
1780 END
1790 '

```

カラー・パレットの初期化プログラム 1面モードと2面モード用の初期化ができます。

```

10 ' name "PALET INIT"
1000 '
1010 ' PALLET INITIALIZE PROGRAM
1020 '   Programmed by Joe Masumura
1030 '
1040 DEFINT I-N ' 変数宣言
1050 DEFSNG R,E
1060 DEFDBL D,A
1070 DEFSTR C,H
1080 OPTION BASE 0
1090 '
1100 ' ヒヨウシゝ モートゝ ノ セツテイ
1110 '   1 : 4096 ショク 1 カゝメン モートゝ
1120 '   2 :   64 ショク 2 カゝメン モートゝ
1130 '
1140 CLS
1150 PRINT "1: 4096 color  2: 64 color"
1160 INPUT CMODE
1170 IF CMODE="1" THEN "L4096"
1180 IF CMODE="2" THEN "L64"
1190 GO TO 1140
1200 '
1210 LABEL "L4096" ' 4 0 9 6 色, 1 画面モード
1220 '
1230 ' GRAPHIC INITIALIZE
1240 '
1250 OPTION SCREEN 0
1260 WIDTH 40,25,0,1
1270 OUT &H1FB0,&H80
1280 OUT &H1FC1,&H28
1290 OUT &H1FC2,&H0
1300 OUT &H1FC5,&H80
1310 '
1320 ' ハゝレット ノ ショキカ
1330 '
1340 LOCATE 0,0:PRINT "INITIALIZING"
1350 FOR IG=0 TO 15
1360   LOCATE 0,1:PRINT 15-IG
1370   FOR IR=0 TO 15
1380     FOR IB=0 TO 15
1390       IAD=&H1000+IG*16+IR
1400       IDT=IB*16
1410       '
1420       OUT IAD,IDT+IB
1430       IAD=IAD+&H100
1440       OUT IAD,IDT+IR
1450       IAD=IAD+&H100
1460       OUT IAD,IDT+IG
1470     NEXT
1480   NEXT
1490 NEXT
1500 '
1510 ' ハゝレット ノ ショキカ END
1520 '

```

```

1530 OUT &H1FC5,&H0
1540 END
1550 '
1560 LABEL "L64" ' 64色, 2画面モード
1570 '
1580 ' GRAPHIC INITIALIZE
1590 '
1600 OPTION SCREEN 0
1610 WIDTH 40,25,0,1
1620 OUT &H1FB0,&H80
1630 OUT &H1FC1,&H28
1640 OUT &H1FC2,&H80
1650 OUT &H1FC5,&H80
1660 '
1670 ' スクリーン 1 ノ ハレット ノ ショキカ
1680 '
1690 LOCATE 0,0:PRINT "SCREEN 1 INITIALIZING"
1700 FOR IG=0 TO 3
1710 LOCATE 0,1:PRINT 3-IG
1720 FOR IR=0 TO 3
1730 FOR IB=0 TO 3
1740 IAD=&H1000+IG*64+IR*4
1750 IDT=IB*64
1760 '
1770 OUT IAD,IDT+IB*4
1780 IAD=IAD+&H100
1790 OUT IAD,IDT+IR*4
1800 IAD=IAD+&H100
1810 OUT IAD,IDT+IG*4
1820 NEXT
1830 NEXT
1840 NEXT
1850 '
1860 ' スクリーン 2 ノ ハレット ノ ショキカ
1870 '
1880 LOCATE 0,2:PRINT "SCREEN 2 INITIALIZING"
1890 FOR IG=0 TO 3
1900 LOCATE 0,3:PRINT 3-IG
1910 FOR IR=0 TO 3
1920 FOR IB=0 TO 3
1930 IAD=&H1000+IG*16+IR
1940 IDT=IB*16
1950 '
1960 OUT IAD,IDT+IB*4
1970 IAD=IAD+&H100
1980 OUT IAD,IDT+IR*4
1990 IAD=IAD+&H100
2000 OUT IAD,IDT+IG*4
2010 NEXT
2020 NEXT
2030 NEXT
2040 '
2050 ' ハレット ノ ショキカ END
2060 '
2070 OUT &H1FC5,&H0
2080 OUT &H1FB0,&H90
2090 END
2100 '

```

1面モードと2面モードのTV画像取り込み例

このプログラムを実行するときは、パレットの初期化が必要です。

```

10 ' name "SCREEN"
1000 '
1010 ' SCREEN TEST PROGRAM
1020 ' Programmed by Joe Masumura
1030 '
1040 DEFINT I-N 変数宣言
1050 DEF$NG R,E

```



```
1060 DEFDBL D,A
1070 DEFSTR C,H
1080 OPTION BASE 0
1090 '
1100 WHILE CMODE<>"Q" ' モード選択
1110   CLS:CMODE="1"
1120   PRINT "1:4096*1 2:64*2 Q:QUIT"
1130   INPUT CMODE
1140   IF CMODE="1" THEN GOSUB "L4096"
1150   IF CMODE="2" THEN GOSUB "L64"
1160 WEND
1170 '
1180 WIDTH 80,25,0,0
1190 '
1200 END
1210 '
1220 LABEL "L4096" ' 4096色, 1画面モード
1230   OPTION SCREEN 0
1240   WIDTH 40,25,0,1
1250   KLIST 0
1260   SCREEN 0,0
1270   OUT &H1FB0,&H80
1280   OUT &H1FC1,&H28
1290   OUT &H1FC2,&H0
1300 '
1310   WHILE CMODE<>"Q" ' 入力処理
1320     CLS:CMODE="G"
1330     PRINT "G:GET Q:QUIT"
1340     INPUT CMODE
1350     IF CMODE="G" THEN GOSUB "GET4096"
1360   WEND
1370 RETURN
1380 '
1390 LABEL "L64" ' 64色, 2画面モード
1400   OPTION SCREEN 0
1410   WIDTH 40,25,0,1
1420   KLIST 0
1430   SCREEN 0,0
1440   OUT &H1FB0,&H90
1450   OUT &H1FC1,&H28
1460   OUT &H1FC2,&H80
1470 '
1480   WHILE CMODE<>"Q" ' 入力処理
1490     CLS:CMODE="G"
1500     PRINT "1:SCREEN1 2:SCREEN2 G:GET Q:QUIT"
1510     INPUT CMODE
1520     IF CMODE="G" THEN GOSUB "GET64"
1530     IF CMODE="1" THEN SCREEN 0,0
1540     IF CMODE="2" THEN SCREEN 2,2
1550   WEND
1560 RETURN
1570 '
1580 ' GET SCREEN 4096 COLOR
1590 '
1600 LABEL "GET4096" ' 4096色時のTV画面取り込み
1610   OUT &H1FB0,&H88
1620   PRINT "HIT ANY KEY"
1630   IF INKEY$="" THEN 1630
1640   OUT &H1FB0,&H80
1650 RETURN
1660 '
1670 ' GET SCREEN 64 COLOR
1680 '
1690 LABEL "GET64" ' 64色時のTV画面取り込み
1700   OUT &H1FB0,&H98
1710   PRINT "HIT ANY KEY"
1720   IF INKEY$="" THEN 1720
1730   OUT &H1FB0,&H90
1740 RETURN
1750 '

```

V-RAMのアドレス計算プログラム

```

10 ' name "VRAM"
1000 '
1010 ' V-RAM ADDRESS PROGRAM
1020 '   Programmed by Joe Masumura
1030 '
1040 ' VARIABLE INITIALIZE
1050 '
1060 '   IXMIN : カ`メン サ`ヒョウ ノ X ノ サイショウチ
1070 '   IXMAX : カ`メン サ`ヒョウ ノ X ノ サイタ`イチ
1080 '   IYMIN : カ`メン サ`ヒョウ ノ Y ノ サイショウチ
1090 '   IYMAX : カ`メン サ`ヒョウ ノ Y ノ サイタ`イチ
1100 '   IPAGE : カ`メン ノ ペ`ージ (ハ`ンク NO.)
1110 '   IX   : アト`レス ケイサン スル テン ノ X サ`ヒョウチ
1120 '   IY   : アト`レス ケイサン スル テン ノ Y サ`ヒョウチ
1130 '
1140 DEFINT I-N
1150 DEFSNG R,E
1160 DEFDBL D,A
1170 DEFSTR C,H
1180 OPTION BASE 0
1190 '
1200 GOSUB "LMODE" ' 解像度の入力
1210 '
1220 GOSUB "LXY" ' X, Y 座標値の入力
1230 '
1240 GOSUB "LVRAM" ' アドレス計算
1250 '
1260 PRINT "BLUE ADDRESS = ";HEX$(IADB)
1270 PRINT "RED ADDRESS = ";HEX$(IADR)
1280 PRINT "GREEN ADDRESS = ";HEX$(IADG)
1290 '
1300 END
1310 '
1320 ' カ`メン ノ カイソ`ウト` ノ ニュウリョク
1330 '
1340 LABEL "LMODE"
1350 CLS:CMODE="0"
1360 PRINT "0:320*200(PAGE0) 1:320*200(PAGE1) 2:640*400"
1370 INPUT CMODE
1380 '
1390 IF CMODE="0" THEN 1400 ELSE 1440
1400 IXMIN=0 : IXMAX=319
1410 IYMIN=0 : IYMAX=199
1420 IPAGE=0
1430 GO TO 1560
1440 IF CMODE="1" THEN 1450 ELSE 1490
1450 IXMIN=0 : IXMAX=319
1460 IYMIN=0 : IYMAX=199
1470 IPAGE=1
1480 GO TO 1560
1490 IF CMODE="2" THEN 1500 ELSE 1540
1500 IXMIN=0 : IXMAX=639
1510 IYMIN=0 : IYMAX=399
1520 IPAGE=0
1530 GO TO 1560
1540 REM OTHER
1550 GO TO "LMODE"
1560 REM END
1570 RETURN
1580 '
1590 ' X,Y サ`ヒョウ ノ ニュウリョク
1600 '
1610 LABEL "LXY"
1620 CLS
1630 PRINT "ENTER X,Y"
1640 INPUT IX,IY
1650 '
1660 IF IX<IXMIN AND IX>IXMAX THEN "LXY"
1670 IF IY<IYMIN AND IY>IYMAX THEN "LXY"
1680 '
1690 CLS
1700 PRINT "X =";IX

```

```

1710 PRINT "Y =";IY
1720 RETURN
1730 '
1740 ' VRAM アドレス ノケイサン
1750 '
1760 LABEL "LVVRAM"
1770 IF IYMAX=199 THEN 1780 ELSE 1860
1780 IF IXMAX=319 THEN 1790 ELSE 1820 ' 200LINEモード
1790 IAD=IPAGE*8H400+(IY%8)*40+IX%8+(IY MOD 8)*8H800
1800 PRINT "320*200 MODE"
1810 GO TO 1840
1820 IAD=IPAGE*8H400+(IY%8)*80+IX%8+(IY MOD 8)*8H800
1830 PRINT "640*200 MODE"
1840 REM END
1850 GO TO 1930
1860 IF IXMAX=319 THEN 1870 ELSE 1900 ' 400LINEモード
1870 IAD=IPAGE*8H400+(IY%16)*40+IX%8+(IY MOD 16)*8H400
1880 PRINT "320*400 MODE"
1890 GO TO 1920
1900 IAD=IPAGE*8H400+(IY%16)*80+IX%8+(IY MOD 16)*8H400
1910 PRINT "640*400 MODE"
1920 REM END
1930 REM END
1940 '
1950 IADB=8H4000+IAD
1960 IADR=8H8000+IAD
1970 IADG=8HC000+IAD
1980 RETURN
1990

```

FM音源のキャリアとモジュレータの波形表示プログラム

```

10 ' name "FM"
1000 '
1010 ' FREQUENCY MODULATION SIMULATION PROGRAM
1020 '   Programmed by Joe Masumura
1030 '
1031 ' VALIABLE INITIALIZE
1032 '
1040 DEFINT I-N ' 変数宣言
1050 DEFSNG R,E
1060 DEFDBL D,A
1070 DEFSTR C,H
1071 OPTION BASE 0
1080 '
1090 ' GRAPHIC INITIALIZE
1100 '
1110 KLIST 0
1120 OPTION SCREEN 0
1130 WIDTH 80,25,1,2
1140 SCREEN 0,0
1150 WINDOW (0,0)-(639,399),(-50,-220)-(419,150)
1160 CLS 4
1170 '
1180 ' ENTER PARAMETER
1190 '
1200 INPUT "CARRIER FREQ RATIO (0.5-95) ";RFC ' キャリアの周波数
1210 INPUT "MODURATOR FREQ RATIO (0.5-95) ";RFM ' モジュレータの周波数
1220 INPUT "CARRIER OUTPUT LEVEL (1-127) ";RAC ' キャリアのレベル
1230 INPUT "MODURATOR OUTPUT LEVEL (0-127) ";RAM ' モジュレータのレベル
1240 '
1250 IF RFC<.5 OR RFC>95 THEN RFC=1 ' エラーチェック
1260 IF RFM<.5 OR RFM>95 THEN RFM=1
1270 IF RAC<1 OR RAC>127 THEN RAC=127
1280 IF RAM<0 OR RAM>127 THEN RAM=127
1290 '
1300 CLS
1310 PRINT "CARRIER FREQ RATIO = ";RFC
1320 PRINT "MODULATOR FREQ RATIO = ";RFM
1330 PRINT "CARRIER OUTPUT LEVEL = ";RAC
1340 PRINT "MODULATOR OUTPUT LEVEL = ";RAM

```

```

1350 '
1360 ' SCALE DISPLAY
1370 '
1380 LINE (-30,0)-(400,0) ' 目盛りの表示
1390 LINE (0,-148)-(0,148)
1400 FOR I=-128 TO 128 STEP 32
1410 LINE (-2,I)-(2,I)
1420 NEXT
1430 FOR I=-112 TO 112 STEP 32
1440 LINE (-1,I)-(1,I)
1450 NEXT
1460 '
1470 ' WAVE DISPLAY
1480 '
1490 R1=3.141592/180 ' 波形の表示
1500 FOR I=0 TO 360
1510 RX=I
1520 RYC=RAC*SIN(RFC*I*R1)
1530 RYM=RAM*SIN(RFM*I*R1)
1540 RYS=RAC*SIN(RFC*I*R1+RYM/RAC)
1550 LINE (RXP,-RYMP)-(RX,-RYM),PSET,1 ' MODULATOR
1560 LINE (RXP,-RYCP)-(RX,-RYC),PSET,5 ' CARRIER
1570 LINE (RXP,-RYSP)-(RX,-RYS),PSET,7
1580 RXP=RX : RYCP=RYC : RYMP=RYM : RYSP=RYS
1590 NEXT

```

グラデーション付き直線の表示プログラム (グラフィックサブルーチンの使用例)

```

10 ' name "LINE"
1000 '
1010 ' LINE DISPLAY PROGRAM
1020 '   Programmed by Joe Masumura
1030 '
1040 ' VARIABLE INITIALIZE
1050 '
1060 '   IX1  : シテン ノ X サ`ヒョウチ
1070 '   IY1  : シテン ノ Y サ`ヒョウチ
1080 '   IX2  : シュウテン ノ X サ`ヒョウチ
1090 '   IY2  : シュウテン ノ Y サ`ヒョウチ
1100 '   IC*  : イロ (R,G,B)
1110 '
1120 DEFINT I-N ' 変数宣言
1130 DEFSNG R,E
1140 DEFDBL D,A
1150 DEFSTR C,H
1160 OPTION BASE 0
1170 '
1180 DIM IC1(2) ' シテン ノ イロ
1190 DIM IC2(2) ' シュウテン ノ イロ
1200 '
1210 ' GRAPHIC INITIALIZE
1220 '
1230 GOSUB "GINIT" ' 初期化サブルーチン
1240 '
1250 ' X1,Y1 カラ X2,Y2 へ セン ラ エカ`ク
1260 '
1270 CLS
1280 PRINT "ENTER X1,Y1,X2,Y2,R1,G1,B1,R2,G2,B2" ' データ入力
1290 INPUT IX1,IY1,IX2,IY2,IC1(0),IC1(1),IC1(2),IC2(0),IC2(1),IC2(2)
1300 '
1310 IF IX1<NXMIN AND IX1>NXMAX THEN 1270 ' エラーチェック
1320 IF IY1<NYMIN AND IY1>NYMAX THEN 1270
1330 IF IX2<NXMIN AND IX2>NXMAX THEN 1270
1340 IF IY2<NYMIN AND IY2>NYMAX THEN 1270
1350 IF IC1(0)<0 AND IC1(0)>15 THEN 1270
1360 IF IC1(1)<0 AND IC1(1)>15 THEN 1270
1370 IF IC1(2)<0 AND IC1(2)>15 THEN 1270
1380 IF IC2(0)<0 AND IC2(0)>15 THEN 1270
1390 IF IC2(1)<0 AND IC2(1)>15 THEN 1270

```

```

1400 IF IC2(2)<0 AND IC2(2)>15 THEN 1270
1410 '
1420 FOR I=0 TO 2 ' 引き数の設定
1430   NC1(I)=IC1(I)
1440   NC2(I)=IC2(I)
1450 NEXT
1460 '
1470 NX1=IX1:NY1=IY1:NX2=IX2:NY2=IY2
1480 '
1490 GOSUB "GLINEG" ' 表示サブルーチン
1500 '
1510 END
1520 '

```

グラデーション付き三角面の表示プログラム (グラフィックサブルーチンの使用例)

```

10 ' name "TRIANGLE"
1000 '
1010 ' TRIANGLE DISPLAY PROGRAM
1020 '   Programmed by Joe Masumura
1030 '
1040 ' VARIABLE INITIALIZE
1050 '
1060 DEFINT I-N
1070 DEFSNG R,E
1080 DEFDBL D,A
1090 DEFSTR C,H
1100 OPTION BASE 0
1110 '
1120 ' GRAPHIC INITIALIZE
1130 '
1140 GOSUB "GINIT" ' 初期化サブルーチン
1150 '
1160 ' X1,Y1 から X2,Y2 へ セン ラ エカク
1170 '
1180 PRINT "X1,Y1,R1,G1,B1" ' P 1 データ入力
1190 INPUT NXX1,NYY1,NCC1(1),NCC1(2),NCC1(0)
1200 '
1210 IF NXX1<NXMIN AND NXX1>NXMAX THEN 1180 ' エラーチェック
1220 IF NYY1<NYMIN AND NYY1>NYMAX THEN 1180
1230 IF NCC1(0)<0 AND NCC1(0)>15 THEN 1180
1240 IF NCC1(1)<0 AND NCC1(1)>15 THEN 1180
1250 IF NCC1(2)<0 AND NCC1(2)>15 THEN 1180
1260 '
1270 PRINT "X2,Y2,R2,G2,B2" ' P 2 データ入力
1280 INPUT NXX2,NYY2,NCC2(1),NCC2(2),NCC2(0)
1290 '
1300 IF NXX2<NXMIN AND NXX2>NXMAX THEN 1270 ' エラーチェック
1310 IF NYY2<NYMIN AND NYY2>NYMAX THEN 1270
1320 IF NCC2(0)<0 AND NCC2(0)>15 THEN 1270
1330 IF NCC2(1)<0 AND NCC2(1)>15 THEN 1270
1340 IF NCC2(2)<0 AND NCC2(2)>15 THEN 1270
1350 '
1360 PRINT "X3,Y3,R3,G3,B3" ' P 3 データ入力
1370 INPUT NXX3,NYY3,NCC3(1),NCC3(2),NCC3(0)
1380 '
1390 IF NXX3<NXMIN AND NXX3>NXMAX THEN 1360 ' エラーチェック
1400 IF NYY3<NYMIN AND NYY3>NYMAX THEN 1360
1410 IF NCC3(0)<0 AND NCC3(0)>15 THEN 1360
1420 IF NCC3(1)<0 AND NCC3(1)>15 THEN 1360
1430 IF NCC3(2)<0 AND NCC3(2)>15 THEN 1360
1440 '
1450 GOSUB "GTRI" ' 三角面の表示サブルーチン
1460 '
1470 END
1480 '

```

4096色を使って点を描くサブルーチン OUT命令を使ったとき

```

12000 REM "GPSET",A ' 12000-
12010 '
12020 ' POINT SET SUBROUTINE
12030 '   Programmed by Joe Masumura
12040 '
12050 '   NX      : X サ`ヒョウチ
12060 '   NY      : Y サ`ヒョウチ
12070 '   NC(0)  : R INTENSITY (0-15)
12080 '   NC(1)  : G INTENSITY (0-15)
12090 '   NC(2)  : B INTENSITY (0-15)
12100 '
12110 LABEL "GPSET"
12120   NADR2=(NY*8)*40+NX*8+(NY AND 7)*8H800
12130 '
12140   NDAT1=INP(8H1FD0) AND 8B11101111 ' RESET
12150   OUT 8H1FD0,NDAT1 ' BANK 0
12160 '
12170   FOR N1=0 TO 2
12180     FOR N2=0 TO 1
12190       NPAGE=N2 AND 1
12200       NADRS=NADR1(N1) OR (NPAGE*8H400+NADR2)
12210       IF (NC(N1) AND 2^N2)=0 THEN 12220 ELSE 12240
12220       NDATA=INP(NADRS) AND ((2^(7-(NX AND 7))) XOR 8HFF) ' RESET
12230       GO TO 12250
12240       NDATA=INP(NADRS) OR (2^(7-(NX AND 7))) ' SET
12250       REM END
12260       OUT NADRS,NDATA
12270       ' PRINT "ADDRESS = ";HEX$(NADRS),
12280       ' PRINT "NDATA  = ";HEX$(NDATA)
12290     NEXT
12300   NEXT
12310 '
12320   NDAT2=NDAT1 OR 8B10000 ' SET
12330   OUT 8H1FD0,NDAT2 ' BANK 1
12340 '
12350   FOR N1=0 TO 2
12360     FOR N2=2 TO 3
12370       NPAGE=N2 AND 1
12380       NADRS=NADR1(N1) OR (NPAGE*8H400+NADR2)
12390       IF (NC(N1) AND 2^N2)=0 THEN 12400 ELSE 12420
12400       NDATA=INP(NADRS) AND ((2^(7-(NX AND 7))) XOR 8HFF) ' RESET
12410       GO TO 12430
12420       NDATA=INP(NADRS) OR (2^(7-(NX AND 7))) ' SET
12430       REM END
12440       OUT NADRS,NDATA
12450       ' PRINT "ADDRESS = ";HEX$(NADRS);"',
12460       ' PRINT "NDATA  = ";HEX$(NDATA)
12470     NEXT
12480   NEXT
12490 RETURN
12500 '

```

4096色を使って点を描くサブルーチン BASICを使ったとき

```

12000 REM "GPSET1",A ' 12000-
12010 '
12020 ' POINT SET ROUTINE
12030 '   Programmed by Joe Masumura
12040 '
12050 '   NX      : X COORDINATE (0-319)
12060 '   NY      : Y COORDINATE (0-199)
12070 '   NC(0)  : R INTENSITY (0-15)
12080 '   NC(1)  : G INTENSITY (0-15)
12090 '   NC(2)  : B INTENSITY (0-15)
12100 '

```

```

12110 LABEL "GPSET"
12120 SCREEN 0,3
12130 NCOL=(NC(1) AND 1)*4 + (NC(0) AND 1)*2 + (NC(2) AND 1)
12140 PSET (NX,NY,NCOL)
12150 '
12160 SCREEN 0,2
12170 NCOL=(NC(1) AND 2)*2 + (NC(0) AND 2) + (NC(2) AND 2)/2
12180 PSET (NX,NY,NCOL)
12190 '
12200 SCREEN 0,1
12210 NCOL=(NC(1) AND 4) + (NC(0) AND 4)/2 + (NC(2) AND 4)/4
12220 PSET (NX,NY,NCOL)
12230 '
12240 SCREEN 0,0
12250 NCOL=(NC(1) AND 8)/2 + (NC(0) AND 8)/4 + (NC(2) AND 8)/8
12260 PSET (NX,NY,NCOL)
12270 RETURN
12280 '

```

直線の両端点の色を補間して表示するサブルーチン

```

16000 REM "GLINEG",A ' 16000-
16010 '
16020 ' GRADATION LINE SUBROUTINE
16030 '   Programmed by Joe Masumura
16040 '
16050 '   NX1   : シテン ノ X サ`ヒョウチ
16060 '   NY1   : シテン ノ Y サ`ヒョウチ
16070 '   NC1(2) : シテン ノ イロ (R,G,B)
16080 '   NX2   : シュウテン ノ X サ`ヒョウチ
16090 '   NY2   : シュウテン ノ Y サ`ヒョウチ
16100 '   NC2(2) : シュウテン ノ イロ (R,G,B)
16110 '
16120 LABEL "GLINEG"
16130 NX=NX1
16140 NDX=ABS(NX2-NX1)
16150 NSX=SGN(NX2-NX1)
16160 NY=NY1
16170 NDY=ABS(NY2-NY1)
16180 NSY=SGN(NY2-NY1)
16190 '
16200 IF NDX>NDY THEN 16210 ELSE 16460
16210 NE=2*NDY-NDX
16220 FOR N1=0 TO 2
16230   NC(N1)=NC1(N1)
16240   NDC(N1)=ABS(NC2(N1)-NC1(N1))
16250   NSC(N1)=SGN(NC2(N1)-NC1(N1))
16260   NEC(N1)=2*NDC(N1)-NDX
16270 NEXT
16280 FOR N=1 TO NDX
16290   GOSUB "GPSET"
16300   IF NE>=0 THEN 16310 ELSE 16330
16310   NY=NY+NSY
16320   NE=NE-2*NDX
16330   REM END
16340   NE=NE+2*NDY
16350   FOR N1=0 TO 2
16360     WHILE NEC(N1)>=0
16370       NC(N1)=NC(N1)+NSC(N1)
16380       NEC(N1)=NEC(N1)-2*NDX
16390     WEND
16400     NEC(N1)=NEC(N1)+2*NDC(N1)
16410   NEXT
16420   PRINT "EX=";NEC(0);NEC(1);NEC(2)
16430   NX=NX+NSX
16440 NEXT
16450 GO TO 16700
16460 NE=2*NDX-NDY

```

```

16470     FOR N1=0 TO 2
16480         NC(N1)=NC1(N1)
16490         NDC(N1)=ABS(NC2(N1)-NC1(N1))
16500         NSC(N1)=SGN(NC2(N1)-NC1(N1))
16510         NEC(N1)=2*NDC(N1)-NDY
16520     NEXT
16530     FOR N=1 TO NDY
16540         GOSUB "GPSET"
16550         IF NE>=0 THEN 16560 ELSE 16580
16560         NX=NX+NSX
16570         NE=NE-2*NDY
16580     REM END
16590     NE=NE+2*NSY
16600     FOR N1=0 TO 2
16610         WHILE NEC(N1)>=0
16620             NC(N1)=NC(N1)+NSC(N1)
16630             NEC(N1)=NEC(N1)-2*NDY
16640         WEND
16650         NEC(N1)=NEC(N1)+2*NDC(N1)
16660     NEXT
16670     PRINT "EY=";NEC(0);NEC(1);NEC(2)
16680     NY=NY+NSY
16690 NEXT
16700 REM END
16710 RETURN
16720 '

```

三角面の内部の色を線形補間するサブルーチン

```

18000 REM "GTRI",A ' 18000-
18010 '
18020 ' TRIANGLE SUBROUTINE
18030 '   Programmed by Joe Masumura
18040 '
18050 '   NXX1   :   チョウテン 1 ノ X サ`ヒョウチ
18060 '   NYY1   :   チョウテン 1 ノ X サ`ヒョウチ
18070 '   NCC1(2) :   チョウテン 1 ノ キト` (0-15)
18080 '   NXX2   :   チョウテン 2 ノ X サ`ヒョウチ
18090 '   NYY2   :   チョウテン 2 ノ X サ`ヒョウチ
18100 '   NCC2(2) :   チョウテン 2 ノ キト` (0-15)
18110 '   NXX3   :   チョウテン 3 ノ X サ`ヒョウチ
18120 '   NYY3   :   チョウテン 3 ノ X サ`ヒョウチ
18130 '   NCC3(2) :   チョウテン 3 ノ キト` (0-15)
18140 '
18150 LABEL "GTRI"
18160 '
18170 '   Y サ`ヒョウ テ` SORT
18180 '
18190 IF NYY1<=NYY2 THEN 18200 ELSE 18600
18200 IF NYY3<=NYY1 THEN 18210 ELSE 18330
18210     NXT1=NXX3
18220     NYT1=NYY3
18230     NXT2=NXX1
18240     NYT2=NYY1
18250     NXT3=NXX2
18260     NYT3=NYY2
18270     FOR NT1=0 TO 2
18280         NCT1(NT1)=NCC3(NT1)
18290         NCT2(NT1)=NCC1(NT1)
18300         NCT3(NT1)=NCC2(NT1)
18310     NEXT
18320     GOTO 18580
18330 IF NYY3<=NYY2 THEN 18340 ELSE 18460
18340     NXT1=NXX1
18350     NYT1=NYY1
18360     NXT2=NXX3
18370     NYT2=NYY3
18380     NXT3=NXX2

```



```

18390      NYT3=NY Y2
18400      FOR NT1=0 TO 2
18410          NCT1(NT1)=NCC1(NT1)
18420          NCT2(NT1)=NCC3(NT1)
18430          NCT3(NT1)=NCC2(NT1)
18440      NEXT
18450      GOTO 18570
18460      NXT1=NXX1
18470      NYT1=NY Y1
18480      NXT2=NXX2
18490      NYT2=NY Y2
18500      NXT3=NXX3
18510      NYT3=NY Y3
18520      FOR NT1=0 TO 2
18530          NCT1(NT1)=NCC1(NT1)
18540          NCT2(NT1)=NCC2(NT1)
18550          NCT3(NT1)=NCC3(NT1)
18560      NEXT
18570      REM END
18580      REM END
18590      GOTO 18990
18600      IF NY Y3<=NY Y2 THEN 18610 ELSE 18730
18610      NXT1=NXX3
18620      NYT1=NY Y3
18630      NXT2=NXX2
18640      NYT2=NY Y2
18650      NXT3=NXX1
18660      NYT3=NY Y1
18670      FOR NT1=0 TO 2
18680          NCT1(NT1)=NCC3(NT1)
18690          NCT2(NT1)=NCC2(NT1)
18700          NCT3(NT1)=NCC1(NT1)
18710      NEXT
18720      GOTO 18980
18730      IF NY Y3<=NY Y1 THEN 18740 ELSE 18860
18740      NXT1=NXX2
18750      NYT1=NY Y2
18760      NXT2=NXX3
18770      NYT2=NY Y3
18780      NXT3=NXX1
18790      NYT3=NY Y1
18800      FOR NT1=0 TO 2
18810          NCT1(NT1)=NCC2(NT1)
18820          NCT2(NT1)=NCC3(NT1)
18830          NCT3(NT1)=NCC1(NT1)
18840      NEXT
18850      GOTO 18970
18860      NXT1=NXX2
18870      NYT1=NY Y2
18880      NXT2=NXX1
18890      NYT2=NY Y1
18900      NXT3=NXX3
18910      NYT3=NY Y3
18920      FOR NT1=0 TO 2
18930          NCT1(NT1)=NCC2(NT1)
18940          NCT2(NT1)=NCC1(NT1)
18950          NCT3(NT1)=NCC3(NT1)
18960      NEXT
18970      REM END
18980      REM END
18990      REM END
19000      '
19010      PRINT "X1="NXT1,"Y1=";NYT1
19020      PRINT "X2="NXT2,"Y2=";NYT2
19030      PRINT "X3="NXT3,"Y3=";NYT3
19040      '
19050      NX1=NXT1
19060      NY1=NYT1
19070      NDXT1=ABS(NXT3-NXT1)
19080      NDYT1=ABS(NYT3-NYT1)
19090      NSXT1=SGN(NXT3-NXT1)
19100      NSYT1=SGN(NYT3-NYT1)
19110      NET1=2*NDXT1-NDYT1
19120      FOR NT2=0 TO 2

```

```

19130     NC1 (NT2)=NCT1 (NT2)
19140     NDCT1 (NT2)=ABS (NCT3 (NT2)-NCT1 (NT2))
19150     NSCT1 (NT2)=SGN (NCT3 (NT2)-NCT1 (NT2))
19160     NECT1 (NT2)=2*NDCT1 (NT2)-NDXT1 (NT2)
19170     NEXT
19180     '
19190     NX2=NXT1
19200     NY2=NYT1
19210     NDXT2=ABS (NXT2-NXT1)
19220     NDYT2=ABS (NYT2-NYT1)
19230     NSXT2=SGN (NXT2-NXT1)
19240     NSYT2=SGN (NYT2-NYT1)
19250     NET2=2*NDXT2-NDYT2
19260     FOR NT2=0 TO 2
19270         NC2 (NT2)=NCT2 (NT2)
19280         NDCT2 (NT2)=ABS (NCT2 (NT2)-NCT1 (NT2))
19290         NSCT2 (NT2)=SGN (NCT2 (NT2)-NCT1 (NT2))
19300         NECT2 (NT2)=2*NDCT2 (NT2)-NDXT1 (NT2)
19310     NEXT
19320     '
19330     GOSUB "GLINEG"
19340     '
19350     FOR NT1=1 TO NDYT2
19360         WHILE NET1>=0
19370             NX1=NX1+NSXT1
19380             NET1=NET1-2*NDYT1
19390         WEND
19400         NET1=NET1+2*NDXT1
19410         '
19420         FOR NT2=0 TO 2
19430             WHILE NECT1 (NT2)>=0
19440                 NC1 (NT2)=NC1 (NT2)+NSC1 (NT2)
19450                 NECT1 (NT2)=NDCT1 (NT2)-2*NDYT1
19460             WEND
19470         NEXT
19480         '
19490         WHILE NET2>=0
19500             NX2=NX2+NSXT2
19510             NET2=NET2-2*NDYT2
19520         WEND
19530         NET2=NET2+2*NDXT2
19540         '
19550         FOR NT2=0 TO 2
19560             WHILE NECT2 (NT2)>=0
19570                 NC2 (NT2)=NC2 (NT2)+NSC2 (NT2)
19580                 NECT2 (NT2)=NDCT2 (NT2)-2*NDYT2
19590             WEND
19600         NEXT
19610         '
19620         NY1=NY1+NSYT1
19630         NY2=NY2+NSYT2
19640         '
19650         GOSUB "GLINEG"
19660     NEXT
19670     '
19680     RETURN
19690     '

```

グラフィック処理の初期化サブルーチン

```

10000 REM "GINIT",A ' 10000-
10010 '
10020 ' GRAPHIC INITIALIZE SUBROUTINE
10030 '   Programmed by Joe Masumura
10040 '
10050 LABEL "GINIT"
10060 '
10070 '   VARIABLE INITIALIZE
10080 '
10090 DIM NC(2) ' 変数宣言
10100 DIM NC1(2)
10110 DIM NC2(2)
10120 DIM NCC1(2)
10130 DIM NCC2(2)
10140 DIM NCC3(2)
10150 DIM NCT1(2)
10160 DIM NCT2(2)
10170 DIM NCT3(2)
10180 DIM NDC1(2)
10190 DIM NDC2(2)
10200 DIM NDC3(2)
10210 DIM NSC1(2)
10220 DIM NSC2(2)
10230 DIM NSC3(2)
10240 DIM NEC1(2)
10250 DIM NEC2(2)
10260 DIM NEC3(2)
10270 DIM NADR1(2)
10280 '
10290 NXMIN=0 ' 変数の初期化
10300 NYMIN=0
10310 NXMAX=319
10320 NYMAX=199
10330 NPAGE=0
10340 NADR1(0)=8H8000 ' R
10350 NADR1(1)=8HC000 ' G
10360 NADR1(2)=8H4000 ' B
10370 '
10380 ' GRAPHIC INITIALIZE
10390 '
10400 WIDTH 40,25,0,1 ' グラフィック処理の初期化
10410 OPTION SCREEN 0
10420 SCREEN 0,0
10430 KEY LIST 0
10440 OUT &H1FB0,&H80
10450 OUT &H1FC1,&H28
10460 OUT &H1FC2,&H0
10470 RETURN
10480 '

```

画面消去サブルーチン

```

20000 REM "GCLS",A ' 20000-
20010 '
20020 LABEL "GCLS"
20030   FOR NSCRN=3 TO 0 STEP -1
20040     SCREEN 0,NSCRN
20050     CLS 0
20060   NEXT
20070 RETURN
20080 '

```

TV画像取り込みサブルーチン

```
22000 REM "GSCAN",A ' 22000-
22010 '
22020 ' TV DATA SCAN ROUTINE
22030 '
22040 LABEL "GSCAN"
22050 ND=INP(&H1FB0)
22060 ND=ND OR &H8
22070 OUT &H1FB0,ND
22080 '
22090 IF INKEY$="" THEN 22090
22100 '
22110 ND=INP(&H1FB0)
22120 ND=ND AND &HF7
22130 OUT &H1FB0,ND
22140 RETURN
22150 '
```

ストロボ機能サブルーチン

```
24000 REM "GSTROBO",A ' 24000-
24010 '
24020 LABEL "GSTROBO"
24030 PRINT "ENTER TIME"
24040 NJ=200
24050 INPUT NJ
24060 IF NJ<1 OR NJ>1000 THEN NJ=200
24070 NIO=INP(&H1FB0)
24080 NIO=NIO OR &H8
24090 OUT &H1FB0,NIO
24100 '
24110 NIO=INP(&H1FB0)
24120 NIO=NIO AND &HF7
24130 OUT &H1FB0,NIO
24140 FOR I=1 TO NJ
24150 IF INKEY$<>"" THEN 24180
24160 NEXT
24170 GO TO 24070
24180 RETURN
24190 '
```

アート機能サブルーチン 量子化処理による階調数の変更

```
26000 REM "GART",A ' 26000-
26010 '
26020 ' ART CONTROL ROUTINE
26030 ' NP : カイチョウ スウ (0-3)
26040 '
26050 LABEL "GART"
26060 NP=(NP AND 3)*64
26070 ND=INP(&H1FC2) AND &H3F
26080 ND=ND OR NP
26090 OUT &H1FC2,ND
26100 RETURN
26110 '
```

モザイク機能サブルーチン

```
28000 REM "GMOZAIKU",A ' 28000-
28010 '
28020 ' MOZAIKU CONTROL ROUTINE
28030 '   NX : X   ホウコウ ノ オオキサ (0-6)
28040 '   NY : Y   ホウコウ ノ オオキサ (0-5)
28050 '
28060 LABEL "GMOZAIKU"
28070 NP=(NY AND 7)*8+(NX AND 7)
28080 ND=INP(&H1FC2) AND &HCO
28090 ND=ND OR NP
28100 OUT &H1FC2,ND
28110 RETURN
28120 '
```

II部 テクニカル編

第II部をお読みになる前に

第II部ではX1シリーズの技術情報を9章にわけて解説しています。

読者は、必要な情報だけをひろいだすことも、また、順に読み流していくこともできるようになっています。付録としてI/Oマップ、BIOSマップをつけておきましたので、御活用下さい。

なお、本文中に掲載してあるプログラムはturboシリーズ用です。動作させる場合は、BIOSのある領域にプログラムを置かないよう注意して下さい。

第1章

システム概説

X1 が、衝撃的なデビューを飾ったのは昭和57年のことでした。パソコンとテレビの融合という新しいコンセプトは、ユーザーに好意的に迎え入れられ、カラー PCG、PSG といった新しい機能を用いたゲームも続々と登場しました。その後、グラフィック V-RAM や漢字 ROM、3 インチ FD 等のオプションを標準装備するなどのマイナーチェンジを重ね、次第に機能も充実してきました。59年に、X1 に完全な上位互換性を持たせた X1turbo が発売になり、さらに強力なラインアップを揃えました。X1turbo シリーズも turbo II, turbo III と発展し、最新の turboZ に至っています。

1-1 ハードウェア概説

1-1-1 CPU

X1 の CPU には、世界でも最もポピュラーな 8 ビットマイクロプロセッサである「Z-80A」が使われています。Z-80 は、インテル社の 8080 にソフトウェア上位互換性を持っています。Z-80A の「A」は、クロック周波数 4MHz に対応するバージョンであることを示しています(ちなみに、Z-80B は 6MHz、Z-80H は 8MHz、Z-80 は 2.5MHz となっています)。

1-1-2 メモリー構成

X1 には 64K バイトの RAM と IPL 用の ROM の 4K バイトが内蔵されています。turbo シリーズには、それ以外に IPL を含む BIOS ROM として 32K バイトが装備されています。また、画面表示用のグラフィック V-RAM が 48K バイト、turbo シリーズには 96K バイト実装されています。キャラクタ V-RAM も 4K バイト、turbo シリーズにはさらに漢字表示用の漢字 V-RAM が 2K バイト用意されています。V-RAM は I/O ポートに割り当てられているため、バンク切り換えなどの手段をとらずに直接アクセスすることができます。

画面に表示されるキャラクタのドットパターンのデータ(フォント)は CGROM(キャラクタジェネレータ ROM)に記憶されています。X1 シリーズの場合 1 キャラクタは 8×8 ドットで構成されているので、1 つの文字データにつき 64 ビット (8 バイト) 必要とされます。キャラクタは 256 個あるので CGROM は $8 \text{ バイト} \times 256 = 2\text{K}$ バイトの容量を持ちます。X1turbo シリーズの場合は、高解像度モードと低解像度モードの 2 種類の ANK フォントを各 4K バイトずつ計 8K バイト持っています。

また、JIS 第 1 水準漢字 ROM として 128K バイトが実装され、さらにオプションとして、第 2 水準漢字 ROM を 128K バイト持っています。

X1 シリーズには、ユーザーが自由に定義できる文字パターンとして PCG(プログラマブル・キャラクタ・ジェネレータ)が用意されています。この部分は、書き換え可能にするために RAM になっており、1 キャラクタにつき R、G、B 各色 8 バイトずつ、256 キャラクタ分で計 6K バイト用意されています。

1-1-3 サブ CPU

X1 シリーズには、メイン CPU の Z-80A のほかに、入出力用の CPU が 2 個搭載されています。この CPU は、サブ CPU と呼ばれ、ユーザーが直接管理することはできません。メイン CPU が必要に応じて、サブ CPU にコマンドを送り、処理を行います。サブ CPU として、X1 には 80C48 と 80C49 が 1 個ずつ、turbo シリーズには 80C49 が 2 個搭載されています。

サブ CPU のうちの 1 つは、X1 本体内でなくキーボード内に置かれており、押されたキーの情報をシリアルデータに変換して本体に送ります。一方、本体内のサブ CPU は、キーボードから送られてきたキーデータの受け取りや、カセット、テレビ等のコントロール、タイマー回路のコントロールを行います。

サブ CPU は、メイン CPU から完全に独立して仕事をするわけではないので、メイン CPU との間でデータ交信の必要がでてきます。そこで、両 CPU の通信のために 8255 という PPI (Programmable Peripheral Interface) が使われています。また、カセットやプリンターとのインターフェイス用に、もう一個 8255 が使われています。

1-1-4 サウンド機能

X1 シリーズにはサウンド用 LSI として、AY-3-8910 (PSG) が搭載されています。また、FM 音源ボードには YM2151 と呼ばれる LSI が使われています。turboZ で使用されている PSG は YM2149 と呼ばれる AY-3-8910 にソフトウェア的にコンパチブルな LSI です。PSG にはサウンド機能の他に、外部 I/O の機能もあり X1 シリーズではジョイスティックポートとして使われています。また turboZ では、FM 音源が標準で内蔵されています。

1-1-5 画面表示

X1 では、本体にある画面のデータを TV ディスプレイに表示するために HD46505 という CRT コントローラ (CRTC) を採用しています。機能的にはシンプルですが、使いやすく応用が利くので広く普及している LSI です。この CRTC は、水平垂直同期信号の発生、キャラクタとグラフィックの表示アドレスの発生、表示タイミング等をコントロールします。

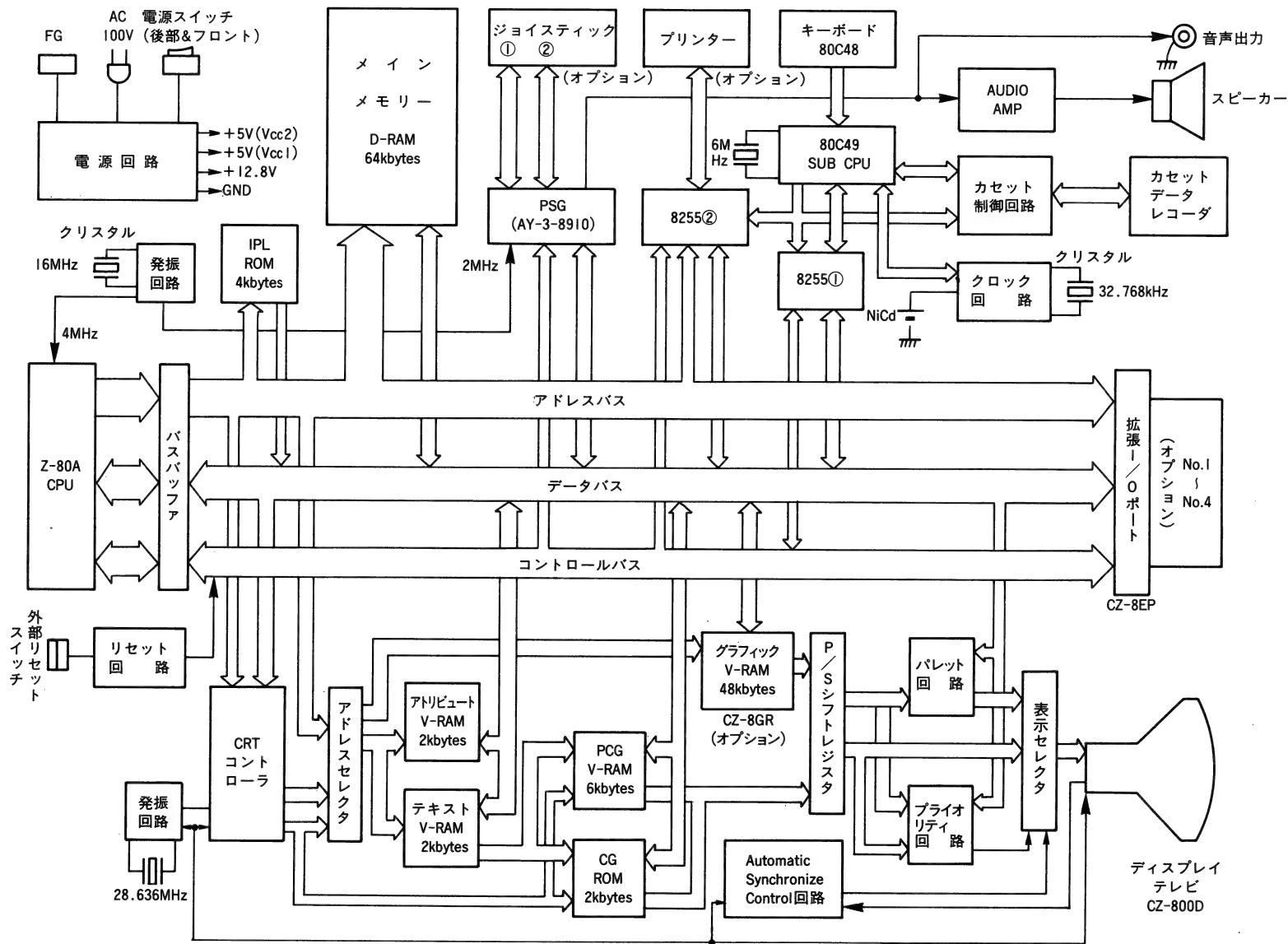
画面のどこにどのような文字が入っているかの情報を示すのがテキスト V-RAM です。80桁×25行=2000字分、2K バイト用意されています (2K=2048ですから48バイト余りますが、この部分は表示されません。この部分は PCG の定義、読み出しに使われています)。また、それぞれの文字がどういう属性 (CGROM か、PCG か、色、大きさなどの性質) を持つのかという情報 (アトリビュート) は、アトリビュート V-RAM 2K バイトに格納されています。turbo シリーズでは、漢字表示用の漢字テキスト V-RAM 2K バイトが別に用意されています。

グラフィック用の V-RAM として、X1 シリーズは 48K バイト、turbo シリーズでは 96K バイトが用意され、解像度を 320×200 ドット、640×200 ドット、turbo シリーズの場合はさらに 320×400 ドット、640×400 ドットと変えることができます。

1-2 X1 シリーズのハードウェア比較

X1 シリーズには多くの機種がありますが、代表として、X1、X1turbo III、X1turboZ のシステムブロック図を示します (図 1-1、1-2、1-3)。システムのメインとなる部分は変更せずに、メモリーの増設や周辺機器の内蔵といった形で進化してきていることがわかります。

図1-1 X1シリーズハードウェアブロック図



1-2 X1シリーズのハードウェア比較

図1-2 X1 turbo IIIシステムブロック図

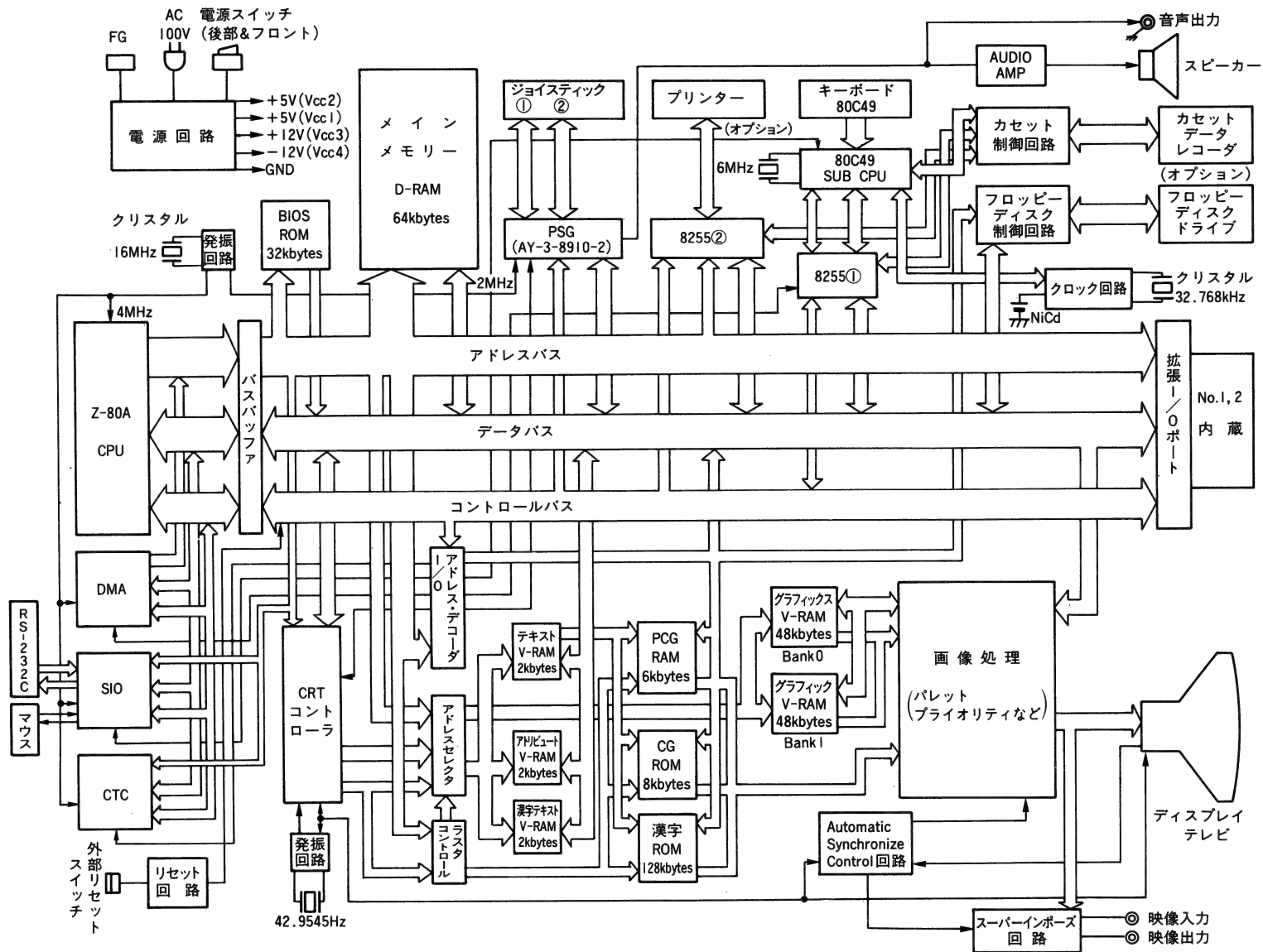
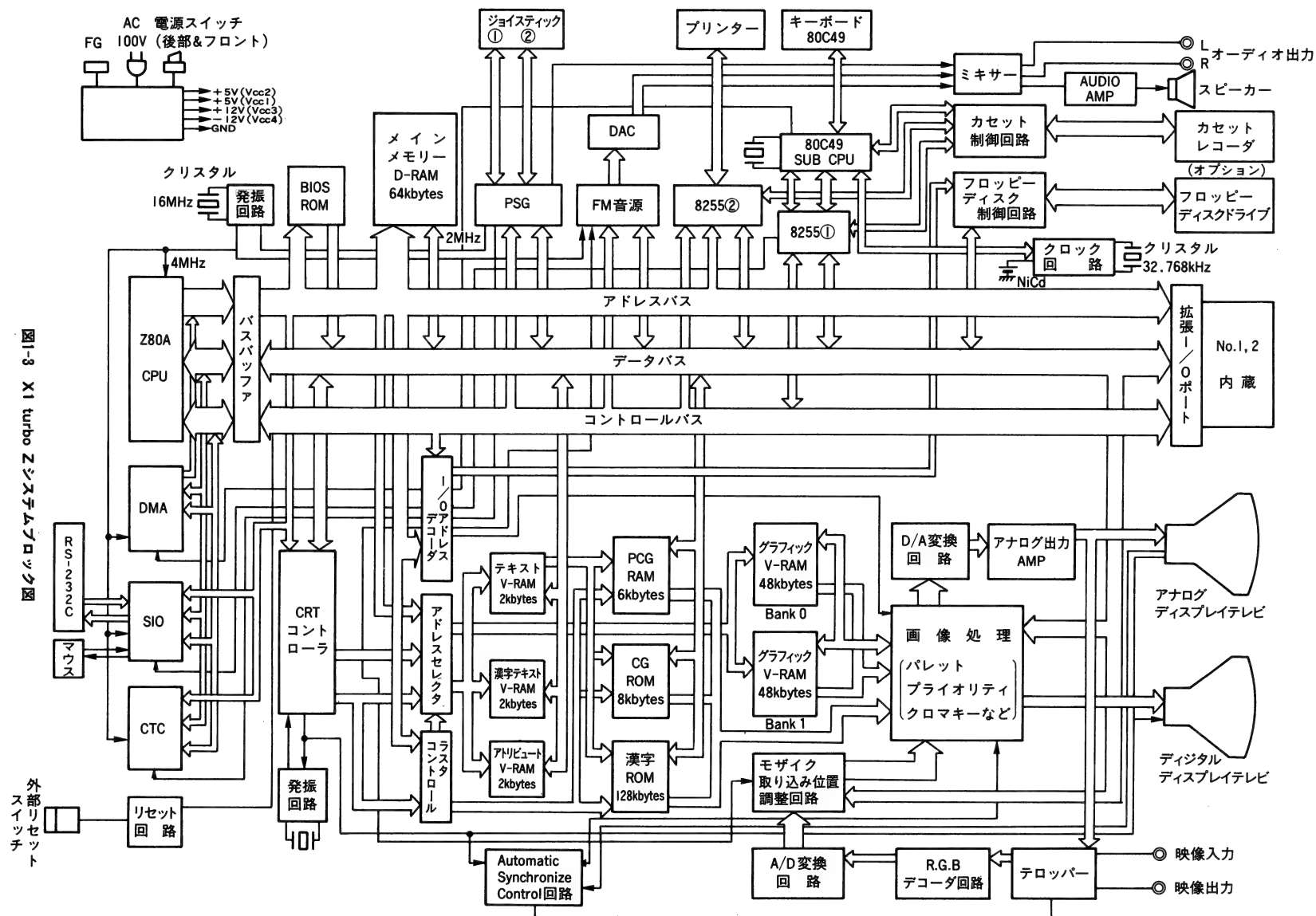


図1-3 X1 turbo Z80システムブロック図



1-2 X1シリーズのハードウェア比較

X1シリーズに標準装備された周辺機器類を表にまとめると次のようになります。

	X1	Cs	Ck	D	F	G	turbo	II	III	Z
GRAPHIC-V-RAM (48K)		○	○	○	○	○				
GRAPHIC-V-RAM (96K)							○	○	○	○
第1水準漢字ROM			○			○	○	○	○	○
第2水準漢字ROM									○	○
漢字テキストRAM							○	○	○	○
カセット	○	○	○		*	*	*			
3" FD				○						
5" 2D					*	*	*	○	○	○
5" 2HD									○	○
RS-232C端子							○	○	○	○
マウス端子							○	○	○	○
FM音源										○

注) ○…標準装備 *…モデルによる

表1-1 X1シリーズに標準装備された周辺機器類

1-3 ソフトウェア概説

X1シリーズには、X1の持つハードウェアの機能をすべてサポートするHuBASICが付属しています(X1turboZを除く)。また、X1F以降の機種には機能を向上させたNEW BASICが付属しています。

X1シリーズのBASICには次の5種類があります。

CZ-8CB01 V1.0 テープ用BASIC

CZ-8FB01 V1.0 ディスク用BASIC

CZ-8CB01 V2.0 テープ用NEW BASIC

CZ-8FB01 V2.0 ディスク用NEW BASIC

○ CZ-8FB02 V1.0 漢字HuBASIC(turbo BASIC)

このうち、漢字HuBASICはturboシリーズでしか使えませんが、他のBASICは全てのX1シリーズで使用することができます。

また、X1のCPUであるZ-80には、CP/M(シーピーエム)というディスクオペレーティングシステム(DOS)があります。このDOSは、8ビットCPU用としては世界で最もポピュラーなものです。DOS上ではプログラムやデータは機種の違いを超えて互換性があります。もちろん、X1シリーズにもランゲージマスター、turboCP/Mの名称で供給されていて、これを使用することにより、英文ワードプロセッサや、事務計算ソフト、FORTRAN、C、LISPといった言語など、CP/M上で動作する多くのソフトウェアが使えるようになります。

第 2 章

メモリー構成

2-1 メインメモリー

2-1-1 X1 シリーズのメモリー構成

X1のメモリー空間は、メインメモリー用のRAM 64KバイトとIPL(Initial Program Loader)ROM 4Kバイトによって構成されています。またX1turbo, X1turboZのメモリー空間は、メインメモリー用のRAM64KバイトとIPL部を含むBIOS ROM32Kバイトによって構成されています。

グラフィック V-RAM, テキスト V-RAM, アトリビュート V-RAM などはI/O 空間 64K バイトの中に配置されています。

2-1-2 増設 RAM

X1で使用されている外部RAM(CZ-8EM)はI/O アドレス(0D ** H)を介してアクセスされる外部記憶装置ですが、turbo シリーズの増設 RAM はメインメモリーの0000H~7FFFH 番地(32K バイト)を1ブロックとする最大16バンク、計512K バイトのメインメモリーとして使用することができます。

メインメモリーと増設 RAM のバンク切り換えは、I/O ポート 0B00H 番地にデータをセットすることによって行われます。

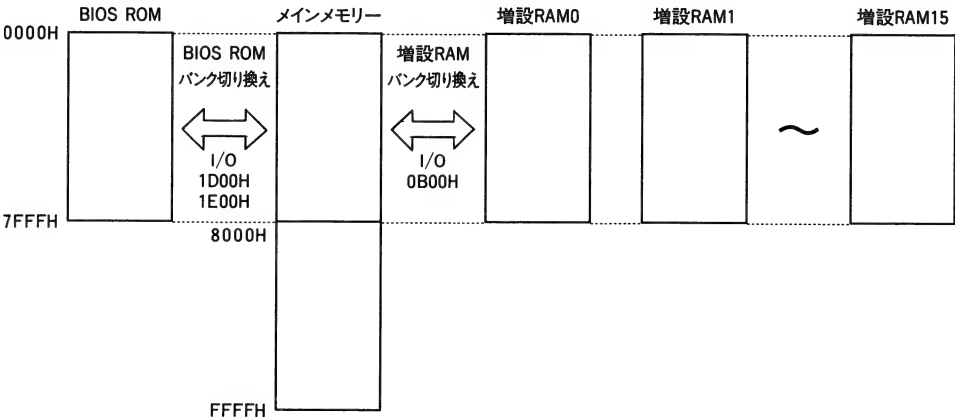
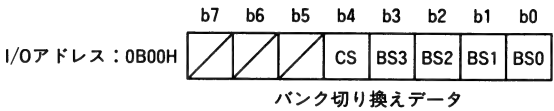


図2-1 増設RAMの切り換え



CS	BS3	BS2	BS1	BS0	バンク切り換え先
1	*	*	*	*	メインメモリー
0	0	0	0	0	増設RAM 0
	0	0	0	1	" 1
	0	0	1	0	" 2
	0	0	1	1	" 3
	0	1	0	0	" 4
	0	1	0	1	" 5
	0	1	1	0	" 6
	0	1	1	1	" 7
	1	0	0	0	" 8
	1	0	0	1	" 9
	1	0	1	0	" 10
	1	0	1	1	" 11
	1	1	0	0	" 12
	1	1	0	1	" 13
	1	1	1	0	" 14
	1	1	1	1	" 15

表2-1 I/Oポート0B00H番地の内容

リスト2-1 増設RAM切り換えプログラム

```
RBKADD EQU 0B00H
RASLAD EQU 1E00H
RBKST: LD BC, RASLAD
      OUT (C), A
      LD BC, RBKADD
      LD A, (RBKDT)
      OUT (C), A
      RET
      ;
RBKDT: DB 01H .....増設RAM#1選択
      ;
      END
```

2-2 内蔵 ROM

2-2-1 IPL ROM

(1) IPL ROM の動作

IPL ROM は電源投入時(IPL リセット時)に時刻やテレビタイマーの設定を行うか、もしくはフロッピーディスク、カセット、ROM 等の外部ソフトウェアをメインメモリーの 0000H 番地以降に読み込みます。

IPL プログラムの実行時には、IPL ROM(0000H 番地から 0FFFH 番地)とメインメモリーの領域は重なっていますが、CPU の読み出し動作時には IPL ROM を、書き込み動作時にはメインメモリーをアクセスするハードウェア構成によって両者を区別しています。IPL ROM は外部

システムプログラムの読み込みが終了するとメインメモリー空間から切り離されます。ただし、タイマー設定時には再起動されます。

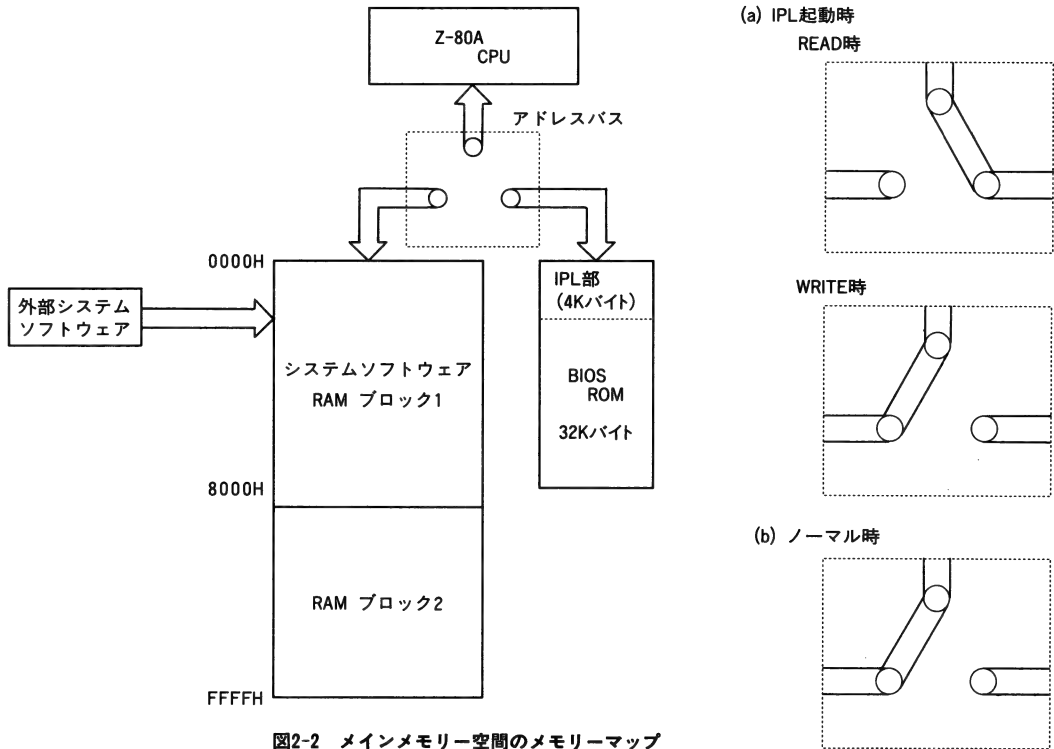


図2-2 メインメモリー空間のメモリーマップ

(2) 外部システムプログラムの読み込み

電源投入時はX1シリーズのメインメモリーが全てRAM構成になっているため、システムソフトウェアを外部デバイスからRAM上に読み込む必要があります。前述したように、システムソフトウェアを読み込むためのプログラムはIPLに組み込まれています。外部デバイスとしてフロッピーディスク、カセット、ROMがサポートされています。

電源投入後IPLが起動して、ハードとソフトの初期設定が行われます。続いてキーが押されているかどうかを調べます。押されたキーの内容と選択するデバイスの内容は以下の通りです。

キーの内容	選択デバイス
F	フロッピーディスク
R	ROM
C	カセット

表2-2 キーの内容と選択デバイス

キーが押されていれば、それに対応するデバイスの読み込みプログラムを起動します。またこのとき「T」が押されていればタイマー設定のプログラムを、「M」が押されていれば機械語モニタを起動します。何もキーが押されていない場合には、各デバイスの接続の状態を次の優先順位で調べます。

1-フロッピーディスク

2-ROM

3-カセット

次に、選択されたデバイスのファイルから、インフォメーションブロックと呼ばれる32バイトを読み込みます。そして、インフォメーションブロックの先頭のモード部分を見て機械語ファイルであれば、システムプログラムをメモリー上にロードします。ロードが終了すると IPL ROM を切り離し、メモリー上のシステムプログラムに制御を渡します。

(3) IPL ルーチンの呼び出し

IPL(BIOS)ROM 内のルーチンを使いたいとき、たとえばタイマーを設定したいときには IPL (BIOS)ROM をアクティブにします。そのためには、システム I/O ポートの 1D ** H ポートを使用します。このポートに対して出力命令を実行すれば IPL(BIOS)ROM がアクティブになります。

IPL(BIOS)ROM をメモリー空間から切り離すためには、システム I/O ポートの 1E ** H ポートを使用します。このポートに対して出力命令を実行すれば IPL(BIOS)ROM はノンアクティブになります。

X1turbo では、次のような手順で IPL の 001BH(RAM)を通じて BIOS ROM 内のルーチンをメインメモリ上のプログラムから CALL することができます。

まず BC レジスタに BIOS のエントリ番地をセットします。

そして RST 001BH を実行します。

LD BC, (アドレス)

RST 001BH

この場合、メインメモリ側のプログラムも IPL の 001BH 番地のルーチンに対応していなければなりません。参考までにメインメモリと IPL のバンク切り換え例を示します。

番地	(メインメモリ)	(IPL)
0018H	CALL 001BH	
001B H	PUSH BC	NOP
001C H	LD B, 1DH	LD B, 1EH
001E H	OUT (C), B	OUT (C), B
0020H	RET	RET

注 意

外部から利用できるサブルーチンのエントリー番地、内容に関しては X1 と turbo シリーズは共通になっています。ただし、X1 と turbo シリーズの IPL を利用した共通プログラムを作成する場合は、次のことに注意する必要があります。X1 では IPL をアクセス中に 1 ウェイトかかりますが、X1turbo ではウェイトがかかりません。これに伴いループカウンターの定数を変更されています。更に、turbo シリーズではメッセージが追加、変更されているためメッセージエリアのアドレスが変更されています。

2-2-2 BIOS ROM

BIOS とは、X1turbo シリーズのハードウェアを直接制御するためのサブルーチンの集まりのことで、CPU と各種 I/O デバイスとの間の入出力インターフェイスの役割をしています。この BIOS は I/O ドライバーだけでなく、浮動小数点パッケージ、グラフィックパッケージ等も含む

強力なものです。各種サブルーチンのエン트리番地や機能などの詳細は付録の「BIOS ROM マップ」を参照してください。

X1turbo シリーズの BIOS ROM は、メモリーアドレス空間の 0000H～7FFFH 番地を占めています。なお、先頭の 4K バイト (0000H 番地から 0FFFH 番地) は IPL になっています。

(1) BIOS ルーチンの呼び出し

BIOS 内のルーチンを呼び出す方法は、0000H～7FFFH 番地にユーザープログラムがあるかないかによって異なります。7FFFH 番地以前にある場合には 2-2-1 (3) の「IPL ルーチンの呼び出し」と同様に行ってください。8000H 番地以降にある場合は、次のような手順を使用します。

```
LD      B, 1DH
OUT     (C), B
CALL    <アドレス>
LD      B, 1EH
OUT     (C), B
```

(2) BIOS ルーチン内部でのエラー発生

BIOS 内のルーチンでエラーが発生した場合は自動的に F83CH 番地の BIOSER にジャンプします。このルーチンは、エラー処理のためのジャンプルーチンなのでユーザープログラム中で BIOS ルーチンを利用する際には、ユーザー側でエラー処理ルーチンを用意して、BIOSER のジャンプテーブルに登録しておく必要があります。なお、その場合はメインメモリーを以下のように設定して、RST 命令で 0008H 番地から実行します。

<番地>	<メインメモリー>	<IPL>
0 0 0 8 H	EX AF, AF'	
0 0 0 9 H	LD A, 1 DH	
0 0 0 B H	OUT (C), A	
0 0 0 D H		EX AF, AF'
0 0 0 E H	どのような内容でも	CALL 7 D 6 C H
0 0 1 1 H	かまいません。	LD B, 1 E H
0 0 1 3 H		OUT (C), B
0 0 1 5 H	RET	
	LD BC, <アドレス>	注) <アドレス> は、エラー処理と
	RST 0 0 0 8 H	ルーチンのエン트리アドレス

注 意

BIOS ROM 内ルーチンでエラーが発生した場合は、キャリーフラグがセットされます。

2-3 I/O 制御

2-3-1 I/O 空間のアクセス

X1 シリーズは 64K バイトの I/O 空間を持ち、Z-80A の入出力命令によって自由にアクセスすることができます。I/O 空間には各種 I/O の他にグラフィック V-RAM も存在しています。

I/O 空間をアクセスするには、Z-80A の I/O 制御命令である IN 命令と OUT 命令を使用します。I/O 制御命令には 2 つの使い方があります。ひとつは C レジスタを使用した入出力命令で以下のように使います。

IN r, (C)

あるいは、

OUT (C), r

この命令を実行することによって BC レジスタの内容がアドレスとして出力されます。すなわち、B レジスタの内容が上位 8 ビットのアドレスに、C レジスタの内容が下位 8 ビットのアドレスになります。これにより 16 ビットのアドレスが指定され、64K バイトの I/O 空間を制御することができます。

もうひとつは入出力命令の第 2 グループを使用したもので、

IN A, (n)

あるいは、

OUT (n), A

を実行します。n には 8 ビットの定数 (0 ~ 255) が入りこれでは、最大 256 個の I/O 空間しかアクセスできませんが、実際は A レジスタの内容が上位 8 ビットに出力されるので、この命令によっても 64K バイトの I/O 空間を制御することができます。

注 意

I/O ポートのなかで I/O アドレスの一部が ** で表現されているものがありますが、これは下位アドレス等がデコードされていないため、この部分はどのような内容でも構いません。

2-3-2 シングルアクセスモードと同時アクセスモード

X1 シリーズの I/O アクセスは、通常のアクセスモードであるシングルアクセスモードとグラフィック V-RAM に対する書き込み時に使用される同時アクセスモードとがあります。

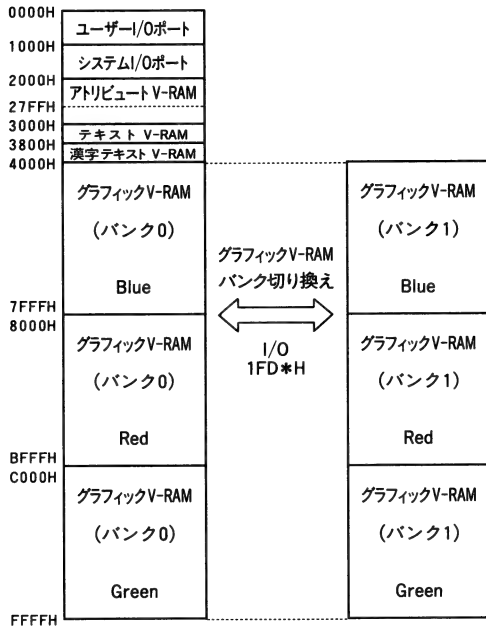
(1) シングルアクセスモード

X1 シリーズではアクセスモードがいかなるモードであっても、I/O に対する読み出し命令を実行するとシングルアクセスモードになるハードウェア構成となっています。従って、シングルアクセスモードで I/O ポートをアクセスするためには、ダミーの IN 命令を実行するだけでアクセス可能となります。

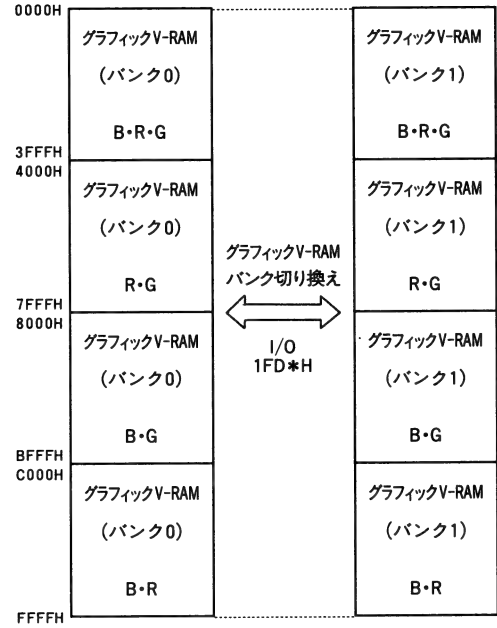
(2) 同時アクセスモード

同時アクセスモードでは、2 面以上のグラフィック V-RAM に同時に書き込むことができるようになっています。このため、面の塗りつぶしや画面のクリアなどを高速に行うことができます。

シングルアクセスモードから同時アクセスモードへ切り換えは、8255②のポート C ビット 5 の立ち下がりによって行われます。そこで、同時アクセスモードにするには、ダミーの読み出し命令を実行してシングルアクセスモードにしたのち、8255②のポート C (1A02H 番地) のビット 5 を "0" → "1" → "0" の順で変化させます。再びシングルアクセスモードにするには、I/O に対して読み出し命令を実行します。



(a) シングルアクセスモード



(b) 同時アクセスモード

図2-3 シングルアクセスモードと同時アクセスモードのI/Oマップ

リスト2-2 シングルアクセスモードでのI/Oポートに対する書き込み

```

SACADD EQU      4000H .....I/Oポートアドレス
SACDAT EQU      20H .....書き込むデータ
SACSS: IN       A, (C) .....シングルアクセスモードに換える
        LD      BC, SACADD
        LD      A, SACDAT
        OUT     (C), A
        RET
        ;
        END

```

リスト2-3 同時アクセスモードでのV-RAMアクセス

```

PIOADD EQU      1A03H .....8255②コントロールレジスタI/Oアドレス
DACADD EQU      4000H .....I/Oアドレス
DACDAT EQU      20H .....データ
DACSS IN       A, (C) .....シングルアクセスモードに
        LD      BC, PIOADD
        LD      A, 0BH
        OUT     (C), A .....8255のポートCのビット5をオン
        DEC     A
        OUT     (C), A .....8255のポートCのビット5をオフ
        LD      BC, DACADD
        LD      A, DACDAT
        OUT     (C), A
        IN      A, (C) .....シングルアクセスへ
        RET
        ;
        END

```

2-3-3 システムI/OポートとユーザーI/Oポート

システムI/OポートとユーザーI/Oポートはシングルアクセスモード時のみアクセスすることができます。

ユーザーI/OポートはI/Oアドレスの0000H～0FFFH番地を占め、各種の周辺機器や外部デバイスなどの入出力に使用します。ただし、シャープから提供されるインターフェイス基板が0100H～0FFFH番地に配置されることになっているので、ユーザーがオリジナルのインターフェイス基板を接続する場合には、0000H～00FFH番地の範囲に設定してください。

システムI/Oは1000H～1FFFH番地に配置されています。このI/Oポートを通じてパレット回路、プライオリティ回路、PCG、CGROM、漢字ROM、CRTC、サブCPU(80C49)、PSG、8255②、IPL-ROMなどの各種デバイスをアクセスしています。システムI/Oポートの詳細は付録「I/Oマップ」を参照してください。

第 3 章

HuBASICの内部構造

原則として、この章はディスク版 NEW BASIC に基づいて書かれています。各 BASIC で異なる場合は、その旨、表記してありますが、基本的には構造は同じです。

3-1 HuBASIC の種類

いままでに X1 に添付されてきた BASIC には次の 5 種類があります。

CZ-8CB01	V1.0	最初のテープ用 BASIC
CZ-8FB01	V1.0	最初のディスク用 BASIC
CZ-8CB01	V2.0	テープ用 NEW BASIC
CZ-8FB01	V2.0	ディスク用 NEW BASIC
<u>CZ-8FB02</u>	V1.0	漢字 HuBASIC (turboBASIC)

このうち、漢字 HuBASIC は X1turbo シリーズでしか使えませんが、他の BASIC はすべての X1 シリーズで起動できます。漢字 HuBASIC は turbo 用の拡張機能をそなえているばかりでなく、高度な日本語処理を行えるようになっています。

NEW BASIC は、フリーエリアの拡張をはかるために、旧 BASIC (V1.0) の中で使用頻度の低かった命令や、他の命令で代用できる命令を削除しました。また、必要とする命令に応じて 10 段階に BASIC の大きさを指定できるなどの特徴を持っています。

3-2 HuBASIC メモリーマップ

HuBASIC(ver1.0) と turbo 版 HuBASIC のメモリーマップを図 3-1 に示します。ただし、ディスク BASIC とテープ BASIC でアドレスが異なる場合にはカッコ内にテープ BASIC の値を示しました。

図中の各エリアについて以下に説明します。

- IOCS (Input Output Control System)

その名称の通りハードウェアに直接関係するデータの出入力を管理します。

- BASIC テキストの先頭アドレス

LIMIT, CLEAR 命令により変更することができます。また、NEW ON レベルに応じて変化します。終了アドレスは BASIC テキストの長さによります。

- 変数エリア

数値変数の場合は種類、変数名と値そのものが入ります。文字変数の場合は種類、変数名、文字列の長さと文字列が格納されているアドレスを知るためのポインタが入っています。

●ファイル用ストリングバッファ

周辺装置とデータをやりとりするときの一時的なバッファ256バイトとデータの入出力先や処理アドレスの入った部分16バイトからなっています。このエリアの大きさは、MAXFILES 命令で指定した最大ファイル数により変化します。最大ファイル数がnのときこのエリアの大きさは $(n + 1) \times (256 + 16)$ バイトとなります。

●ストリングデータエリア

文字変数の実際の文字列データが入るエリアです。

●テンポラリストリングバッファ

文字変数の多重処理、ファイルからの入力時に一時的に使われるエリアです。

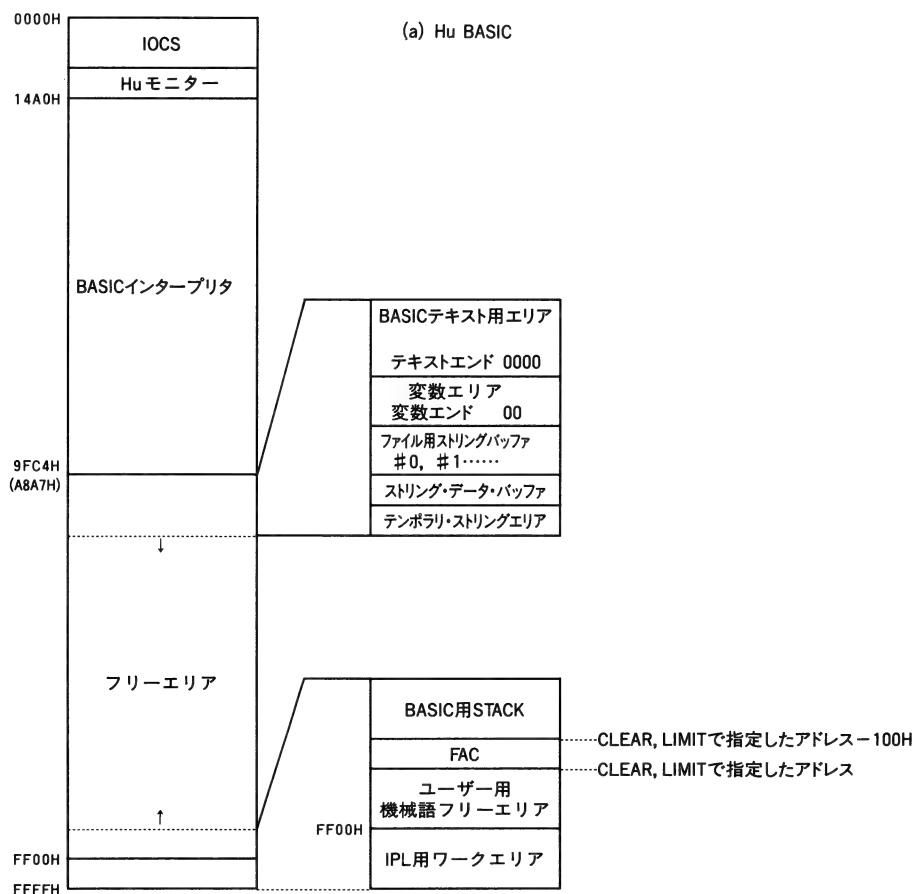
●BASIC 用スタック

GOSUB 文の戻り番地などを保存するエリアです。

●FAC(Floating Accumulator)

FACはBASIC インタープリタのアクムレータで、すべての計算はここを中心に行われます。X1では100H(256バイト)と決められています。USR 関数で渡されたデータや浮動小数点の代入、演算はFACに転送され、FACを経由して処理されます。FACの先頭番地はCLEAR 命令やLIMIT 命令により決定されます。

●FF00H~FFFFH は、汎用のワークエリアです。FF00H からはキー入力バッファやファイルのインフォメーションバッファとして、FFFFH からはスタックとして使われます。



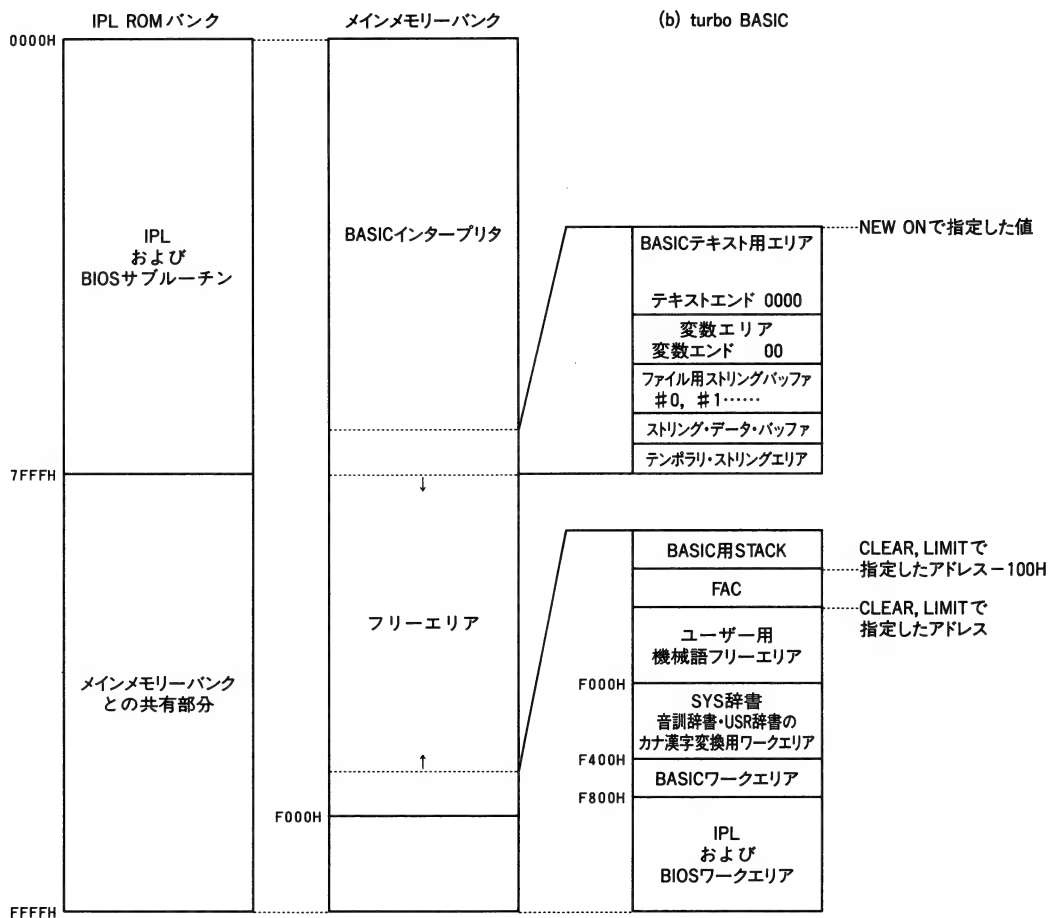


図3-1 Hu BASICメモリーマップ

内 容	格納アドレス カッコ内は DISK BASIC	初 期 値		備 考
		BASIC起動前	起 動 後	
BASICテキスト・トップ	9D52 (A635)	9FC5	9FC5 (A8A8)	NEW ON命令で変えられる
変数エリア・トップ	9D46 (A629)	A0EF	9FC7 (A8AA)	VARPTR命令はこのエリア内のアドレスを示す
ファイル用ストリング・バッファトップ	9D48 (A62B)	A0FC	9FC8 (A8AB)	STRPTR命令で値を見ることができる
ストリング・データ・バッファトップ	35F6 (3628)	A31C	A1E8 (ABDB)	
テンポラリ・ストリングエリア・トップ	9D4A (A62D)	A3D6	A1EA (ABDD)	
フリーエリア・トップ	35EF (3621)	A3D6	A1EA (ABDD)	
FACトップ	9D4C (A62F)	FD00	FE00 (FE00)	CLEAR, LIMITのアドレス－100Hである
ユーザー用機械語フリーエリアトップ	9D4E (A631)	FE00	FF00 (FF00)	CLEAR, LIMIT命令で変えることができる
IPL用ワークエリアトップ	9D50 (A633)	FF00	FF00 (FF00)	半固定

表3-1 HuBASICのワークエリアと初期値

IOCS, インタープリタ部分及び FF00H 以降の IPL などのワークエリアはメモリーマップに示される通りですが, BASIC テキストエリアや変数エリアは状況によって変化します。そのため, 各々の領域の場所を示すポインタがワークエリアの中にあります。このポインタのアドレスと各 BASIC 起動時における初期値を表 3-1 に示します。

また, NEW BASIC における, 各 NEW ON レベルの BASIC テキストスタートアドレスを次表に示します。

NEW ONレベル	0	1	2	3	4	5	6	7	8	9
TEXT先頭番地	7F96	919E	9449	9BB4	9F43	A3A8	A4CC	A587	A6C1	A914

表3-2 BASICテキストスタートアドレス

3-3 プログラムの格納状態

BASIC プログラムは, 行番号順に格納されています。BASIC テキストの格納開始アドレスは, BASIC のバージョンや NEW ON レベルにより変化します。それぞれの行は以下に示すような形式で格納されています。

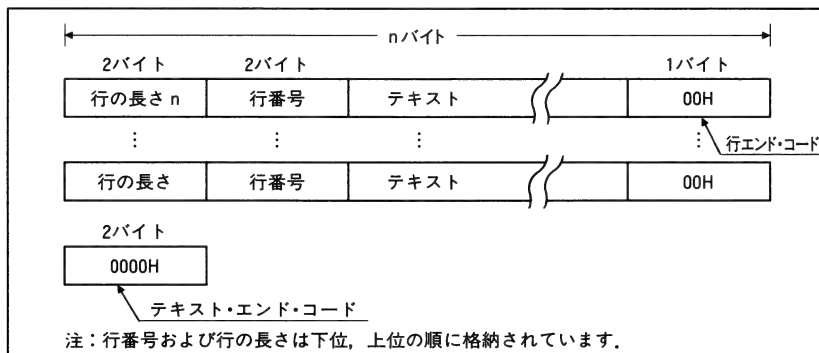


図3-2 テキストの格納形式

プログラムの終りにはテキストエンドコードとして 0000H が書き込まれていて, 後に続く文がないことを示します。

次に, プログラム/テキストがどのように格納されているか見てみましょう。例として次のプログラムを使います。

これを, モニタの D コマンドでダンプしたリストを続いて示します。

リスト3-1

```
10 PRINT "ABCDEF"
20 END
```

```

:A8A8=0F 00 0A 00 8F 20 22 41 /.../ "A
:A8B0=42 43 44 45 46 22 00 06 /BCDEF"..
:A8B8=00 14 00 98 00 00 00 00 /...j....
:A8C0=00 4F 3A 69 00 00 00 00 /.O:i....
:A8C8=00 00 00 00 00 00 00 00 /.....
:A8D0=00 00 00 00 00 00 00 00 /.....
:A8D8=00 00 00 00 00 00 00 00 /.....
:A8E0=00 00 00 00 00 00 00 00 /.....
:A8E8=00 00 00 00 00 00 00 00 /.....
:A8F0=00 00 00 00 00 00 00 00 /.....
:A8F8=00 00 00 00 00 00 00 00 /.....
:A900=00 00 00 00 00 00 00 00 /.....
:A908=00 00 00 00 00 00 00 00 /.....
:A910=00 00 00 00 00 00 00 00 /.....
:A918=00 00 00 00 00 00 00 00 /.....
:A920=00 00 00 00 00 00 00 00 /.....

```

これを見るとわかるように、プログラムは LIST のままの形でメモリーに格納されているわけではなく、できるだけメモリーをつかわないように工夫して格納されています。

3-4 中間言語

前の節で、BASIC プログラムのテキストが圧縮された形で格納されていることがわかりました。これは BASIC のキーワード (PRINT, END など) を 1 バイトか 2 バイトの中間言語で表しているからです。

この PRINT, END といったキーワードは予約語と呼ばれ、BASIC 言語に、あらかじめ登録されています。その一覧は中間言語の順に並べられています。各 BASIC における予約語テーブルアドレスを次に示します。

```

NEW BASIC(DISK)  2 6 7 B H ~ 2 A 0 2 H
旧 BASIC(TAPE)   2 8 F 6 H ~ 2 C D D H
旧 BASIC(DISK)   2 9 2 1 H ~ 2 D 0 B H
turbo 版         5 7 4 1 H ~ 5 B 8 0 H

```

表の構造は、予約語を羅列したものとなっており、各予約語の終りは、次の予約語と区別するために、最後の文字の最上位ビットが ON になっています。また、中間言語テーブルは 3 つに分かれていて、最初が基本的な BASIC の命令、次に拡張された BASIC の命令、最後に関数、システム変数となっています。それぞれの区切りには 0FFH がおかれています。

基本的な BASIC 命令の中間言語は 080H からはじまる 1 バイト、拡張命令は 0FEH+080H から始まる 1 バイトの計 2 バイト、関数は 0FFH+080H からの 1 バイトの計 2 バイトという構成になっています。

3-5 変数テーブル

テキストエリアの後に変数テーブルと呼ばれる、変数の値を格納しておくエリアが用意されています。ただし、プログラムを実行して実際にその変数が使われないと、変数領域に登録されません。

3-5-1 単純変数テーブル

変数の種類、変数名の長さによって必要なバイト数は異なりますが、各変数とも同じような形式で格納されます。文字型変数を除いて、1バイト目で示される変数の種類の値はそのまま変数データの格納バイト数も表しています。

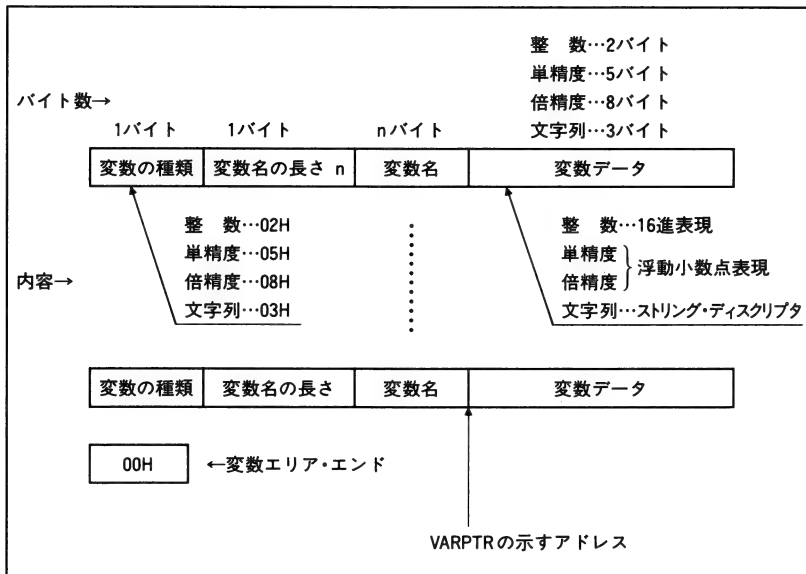


図3-3 変数の格納形式

変数データは、整数型変数の場合は数値がそのまま16進数2バイトで入っており、単精度、倍精度型変数のときには次節で説明する浮動小数点表記で格納されています。また、文字変数の場合は、他の変数型と異なり、直接このエリアにデータは格納されていません。他の変数型のデータ格納部分にあたる部分にはストリングディスクリプタといわれるものが格納されていて、このデータを用いて、実際のストリングデータエリアのアドレスを計算します。ストリングディスクリプタとは文字変数の値(文字列)の格納されているストリングデータエリアの先頭アドレスを0としたとき、何バイト離れたところに文字列の先頭があるかを示す値です。ストリングデータエリアの先頭アドレスはSTRPTR変数に入っているので、実際に文字列が格納されたアドレスは、STRPTR変数の値にストリングディスクリプタを加えることによって求められます。

NEW BASICの場合はSTRPTR変数が削除されていますので、直接BASICの内部ポインタである、7D2AHを参照することによって、ストリングデータエリアの先頭番地を求めて下さい。

では、実際にどのように変数が格納されているかを見てみましょう。次の図は、NEW BASIC DISK版で、NEW ON 9の状態を実験したものです。

リスト3-2

```

10 a%=10
20 a!=10
30 a#=10
40 a$="10"

```

```

:A940=00 00 00 02 01 41 0A 00
:A948=05 01 41 84 20 00 00 00
:A950=08 01 41 84 20 00 00 00
:A958=00 00 00 03 01 41 02 32
:A960=03 00 01 4F 2C 42 B7 28
:A968=27 2A 80 30 E5 1A 13 06
:A970=08 1F CB 16 10 FB 23 0D
:A978=20 F3 22 80 30 EB CD 2E
:A980=56 E1 ED 5B 2A 7D B7 ED
:A988=52 EB E1 E5 23 73 23 72
:A990=E1 C3 65 50 CD 87 A5 F6
:A998=80 FE 80 CA D6 A4 3E 0D
:A9A0=C3 D6 A4 CD 5B 4C 3D FE
:A9A8=0C 30 15 C6 10 C3 D6 A4
:A9B0=3E 03 CA D6 A4 CD 70 4C
:A9B8=1D 7A B7 28 06 3C 28 1D

```

アドレス	データ	意味
A943	02	変数の種類(整数型単純変数)
A944	01	変数名の長さ=1バイト
A945	41	変数名='A'
A946	0A 00	変数データ(0AH=10)
A948	05	変数の種類(単精度型単純変数)
A949	01	変数名の長さ
A94A	41	変数名
A94B	84 20 00 00 00	変数データ(浮動小数点表示)
A950	08	変数の種類(倍精度型単純変数)
A951	01	変数名の長さ
A952	41	変数名
A953	84 20 00 00 00	変数データ
	00 00 00	
A95B	03	変数の種類(文字型単純変数)
A95C	01	変数名の長さ
A95D	41	変数名
A95E	02	変数データの長さ
A95F	03 32	ストリング・ディスクリプタ
A961	00	変数領域の終り

表3-3

3-5-2 配列変数テーブル

配列変数の格納形式は次図のようになっています。



図3-4 配列の格納形式

n次元配列の場合は、「配列の大きさ」の部分がn個になります。また、「配列の大きさ」に格納される順番は、後の引数ほど先になり、変数のデータの格納順序は逆に引数の出てきた順になります。例えば、DIM A(2, 5)と宣言した場合は「配列の大きさ」には引数5が先に格納され、次に引数2が格納されます。変数データ域には、A(0, 0), A(1, 0), A(2, 0), A(0, 1)……A(2, 5)の順に格納されていきます。

変数の種類を示すデータは、単純変数と区別するために、最上位ビットを1にしています。

では、実際にどのように変数が格納されているかを見てみましょう。

(NEW BASIC DISK 版／NEW ON9)

リスト3-3

```
100 DIM a%(1, 2)
110 '
120 a%(0, 1) = 10
130 a%(1, 2) = 12
140 a%(1, 1) = 11
```

```
:A958=00 00 00 82 15 00 01 41
:A960=02 03 00 02 00 00 00 00
:A968=00 0A 00 0B 00 00 00 0C
:A970=00 00 01 4F 2C 42 12 0A
:A978=00 00 0B 00 14 00 61 21
:A980=F4 12 0A 00 00 0B 00 1E
:A988=00 61 23 F4 12 0A 00 00
:A990=0C 00 28 00 61 24 F4 22
:A998=31 30 22 00 00 00 02 01
:A9A0=41 0A 00 05 01 41 84 20
:A9A8=00 00 00 08 01 41 84 20
:A9B0=00 00 00 00 00 00 03 01
:A9B8=41 02 32 03 00 00 4F A9
:A9C0=42 B7 28 27 2A 80 30 E5
:A9C8=1A 13 06 08 1F CB 16 10
:A9D0=FB 23 0D 20 F3 22 80 30
:A9D8=EB CD 2E 56 E1 ED 5B 2A
```

アドレス	データ	意味
A95B	82	変数の種類(整数型配列変数)
A95C	00 15	配列データの終りまでの長さ=15Hバイト
A95E	01	変数名の長さ=1バイト
A95F	41	変数名='A'
A960	02	配列の次元=2次元
A961	00 03	二つ目の引き数+1=3
A963	00 02	一つ目の引き数+1=2
A965	省略	A(0, 0)のデータ A(1, 0) A(0, 1) A(1, 1) A(0, 2) A(1, 2)
A971	00	変数領域の終り

表3-4

3-6 数値の内部表現

3-6-1 浮動小数点

HuBASIC 内部における演算と数値の変数領域への格納は、浮動小数点表記法で行われています。浮動小数点表記では、数値は仮数部+指数部で表現されます。

浮動小数点表記とは、数値を有効数値と位どりの部分にわけて表現する方法で、例えば5430000は 5.43×10^6 と表せます。この方法をとれば、コンピュータ内部に数値を格納する場合にも、少ないバイト数で表現できます。

コンピュータで数値を扱う場合には基本的に2進数ですから、位どり等も2進数で表現します。2進数の浮動小数点表記には、いくつかの方法がありますが、ここではHuBASIC内部で使用する表現を用いて10進の20.5を浮動小数点表記する過程を示してみましょう。

$$\begin{aligned} 20.5 &= 16 + 4 + 0.5 \\ &= 2^4 + 2^2 + 2^{-1} \\ &= 10100.1\text{B} \\ &= 1.01001\text{B} \times 2^4 \end{aligned}$$

0以外の実数はすべてこの形式で表現することができ、数式的に表すと次のようになります。2進数は1か0しかないため、いちばん上の桁(整数部)は常に1となります。

$$\pm 1. \text{*****} \text{B} \times 2^{\pm n} \quad (*\text{は}0\text{か}1)$$

指数部に1バイト割り当て、0の時は指数部を00Hとすれば基本的にすべての数値が表現できます。また、指数部は正負両方の値をとるので、1バイトで表現するために、実際の数値でなく、81Hを0とする指数を用います。この方法では、もとの指数の数値を1とすると82H、2なら83H、同様に-1なら80H、-2なら7FHとなります。

仮数部の表現方法は、その整数部が常に1であるため、このビットを格納する必要はありません。ですから、このビットは仮数部の符号として使用します。仮数部が正の場合は最上位ビットを0、負の時には最上位ビットを1にします。

変数型により仮数部に使用されるバイト数は異なり、単精度数値を格納するためには4バイト、倍精度数値には7バイト使用します。

実際に、先ほどの値20.5($1.01001\text{B} \times 2^4$)を単精度数値としてHuBASICの格納形式で表せば、

$$\begin{array}{ll} \text{指数部} & 4 + 81\text{H} = 85\text{H} \\ \text{仮数部} & 00100100 \quad 00000000 \quad 00000000 \quad 00000000 \\ & = 24\text{H} \quad 00\text{H} \quad 00\text{H} \quad 00\text{H} \end{array}$$

と、なります。

さて、10進数を2進数に変換すると、きれいに割り切れない場合がでてきます。たとえば、0.1は $1.001100110011 \dots \times 2^{-4}$ となって無限循環小数になってしまいます。このような場合は最後の桁を0捨1入します。

$$\begin{array}{ll} \text{指数部} & -4 + 81\text{H} = 7\text{DH} \\ \text{仮数部} & 01001100 \quad 11001100 \quad 11001100 \quad 11001100 \quad 1100\dots \\ & \quad \quad \quad (\uparrow \text{桁上げ}) \\ & = 4\text{CH} \quad \text{CCH} \quad \text{CCH} \quad \text{CDH} \end{array}$$

3-6-2 浮動小数点の演算誤差

浮動小数点表記で、10進数を2進数に変換すると、誤差がでることがあります。前節で、0.1が、無限循環小数になってしまうことを示しましたが、例えば、次のプログラムをHuBASICで実行してみると、

```
A#=0.1
PRINT A#
```

```
0.1000000000058208
```

と表示され、A#は正確に0.1ではないことが確認されます。これは、この場合、仮数部の57ビット目が0捨1入されたためで、このまま2進数で浮動小数点表記を使っている限り、避けることができません。

このような誤差を避けるための方法としては、 10^n 倍して、できるだけ整数の状態で計算をして、表示、登録の段階で 10^n で除して実際の数値に戻す方法などが挙げられます。

3-7 機械語サブルーチンとのリンク

HuBASICには、機械語のサブルーチンを呼び出したい場合のためにCALL命令とUSR関数が用意されています。CALL命令は手軽に使える反面、パラメータの受け渡しが困難という短所があり、逆にUSR関数は関数なのでパラメータの受け渡しは簡単なのですが、少々理解しづらい面があります。

3-7-1 CALL 命令

CALL命令は、直後に書かれたアドレスから始まる機械語サブルーチンを呼び出します。機械語サブルーチンからBASICに復帰するためには機械語プログラムの中でRET命令(0C9H)を実行します。

HuBASICのCALL命令には機械語サブルーチンとのパラメータ受け渡しの機能はサポートされていませんので、PEEK命令、POKE命令を使ってメモリーを介してパラメータを受け渡します。また、マニュアルには書かれていませんが、CALL アドレス (データ)という形式でHLレジスタにデータを渡すこともできます。ただし、機械語サブルーチンからBASICにデータを返す機能はサポートされていないので、データの読みとりはPEEK命令を使います。

なお、機械語サブルーチン内でレジスタを保存する必要はありません。

3-7-2 USR 関数

USR関数は、関数の形で機械語のサブルーチンを呼び出します。機械語サブルーチンから復帰したあとは、その結果を持つ関数になります。USR関数は、実行前に呼び出す機械語サブルーチンの実行開始アドレスを定義する必要がありそれにはDEF USR文を使って次のように書きます。

```
DEF USRn=a
```

nは関数識別番号で、0～9の最大10個の機械語サブルーチンを定義する事ができます。nを省略すると0が指定されます。実行開始アドレスaは、機械語サブルーチンの実行が開始される

アドレスであり、メインメモリー 64KB 中のどこにでも指定できます。

DEF USR 文でいったんアドレスを定義すると、再定義しなすまで、そのアドレスが保持され、NEW や CLEAR 命令を実行しても、消去されません。

DEF USR で、n と a を定義したら次のようにして USR 関数を使うことができます。

- | | |
|-----------------------------|--|
| ① $y = \text{USRn}(x)$ | ① y : 数値変数
n : USR 関数識別番号. 0 ~ 9 の整数.
x : 数式(パラメータ) |
| ② $y \$ = \text{USRn}(x\$)$ | ② $y \$$: 文字変数
n : USR 関数識別番号. 0 ~ 9 の整数.
x \$: 文字式(パラメータ) |

USR の後ろに付ける番号 n は、DEF USR 文で設定した番号に対応します。

USR 関数は、パラメータ x や x\$ の値を持って、n 番の機械語サブルーチンを呼び出し、復帰時に、その値を y や y\$ に代入します。機械語サブルーチンの中で、パラメータの値を書き換えると、その結果が y や y\$ の値となります。USR\$ のパラメータ x、x\$ は省略する事ができません。

x は、数式で、単独の定数もしくは数値変数でもよく、精度も整数型、単精度型、倍精度型のいずれでも構いません。x\$ は、文字列を値に持つ文字式です。

USR 関数で機械語サブルーチンを呼び出すとき CPU の各レジスタには次の表で示される値が入っています。

レジスタ \ パラメータ	数 値 型	文 字 型
A	データの種類・バイト数 02H...整数 05H...単精度 08H...倍精度	03H...文字型のみ
HL	データ格納アドレス	ストリングディスクリプタ 格納アドレス
B	無意味	文字列の長さ
DE	無意味	文字列データ格納先アドレス
IX	エラー処理ルーチン・エントリーアドレス	

表3-5

● A レジスタ(アキュムレータ)

A レジスタ(アキュムレータ)には、パラメータの型を示す値が入っています。これらの値は、文字変数を除き、メモリー内において何バイトで表現されているかを表しています。

● HL レジスタ

HL レジスタの値は、数値変数と文字変数の場合で異なります。数値変数の場合はパラメータの入っているメインメモリーのアドレスを示しており、パラメータが文字変数の場合は、直接文字列の入っているアドレスを示さず、ストリングディスクリプタと呼ばれているテーブルのアドレスを示しています。パラメータの値は、HL レジスタが指しているアドレスから格納されています。

● DE およびBレジスタ

DE とBレジスタは、Aレジスタが3のとき、すなわちUSR関数のパラメータが文字変数の時のみ意味を持ちます。この時DEレジスタにはその文字データが格納されているアドレス(絶対アドレス)、Bレジスタには文字数が入っています。

※ USR関数の引数に文字列を使う場合の注意

USR関数の引数に文字列を使った場合、機械語ルーチンからBASICに制御が戻ったときに、引数に使用した文字列が破壊される場合があります。例えば、A\$を引数に使い、A\$の先頭1文字を削除した文字列を値として返す関数を作った場合、BASICに制御が戻った時点でA\$は先頭1文字が削除されてしまっています。これを避けるためには次のようにします。

DM\$=USR(A\$+"")

この方法をとれば、機械語ルーチンで参照するのは、テンポラリストリングバッファにコピーされたA\$+" " ですから、破壊されるのはそちらであり、A\$は破壊されません。

● USR関数内でのエラー処理

USR関数内でエラーが発生した場合、BASICにエラーの発生を知らせることができます。このとき、BASIC内であらかじめON ERROR GOTOの定義をしていれば、エラー処理ルーチンへジャンプすることもできます。

エラーの通知はエラー番号をAレジスタに入れIXレジスタの示すアドレスにジャンプすることにより行います。

第4章

画面表示

4-1 V-RAM

X1 シリーズは、テキスト系 V-RAM とグラフィック V-RAM の 2 つの系統の V-RAM をもっています。

テキスト画面を構成する「テキスト系 V-RAM」は、テキスト V-RAM、漢字用テキスト V-RAM (turbo シリーズのみ)、アトリビュート V-RAM の 3 種類にわけることができ、それぞれが 2K バイトの容量を持っています。テキスト画面の 1 文字は、各 V-RAM の 1 バイトに対応するので、1 文字は 3 バイト (X1 は 2 バイト) で表されていることになります。この時、V-RAM のアドレスが表示位置に、各 V-RAM に書き込まれている計 3 (2) バイトデータが、表示される文字及び属性に対応します。この 3 (2) バイトのデータが表す内容の概要を以下に示します。

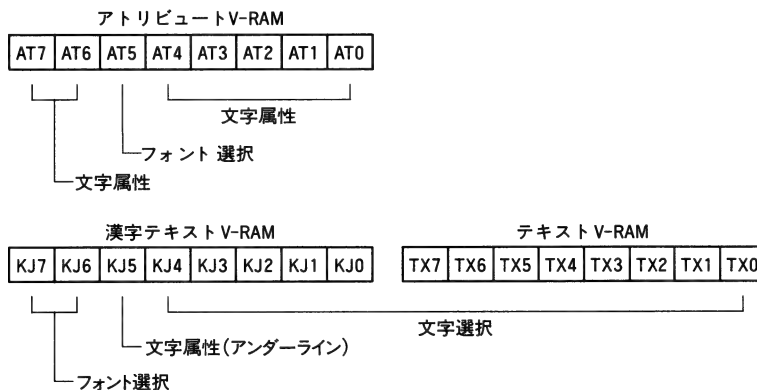


図4-1 テキスト V-RAM のデータが表す内容 (概要)

X1 シリーズでは、この 3 (2) バイトのデータによって、テキスト画面に次の 3 種類の文字のいずれかを表示することができます。

(1) キャラクタ・ジェネレータ (CG)

アルファベット、数字、カタカナ、セミグラフィック等

(2) プログラマブル・キャラクタ・ジェネレータ (PCG)

ユーザー定義可能なキャラクタ・ジェネレータ RAM

(3) 漢字 ROM (turbo シリーズのみ)

JIS 非漢字文字 45 字と第 1 水準漢字 2965 文字、第 2 水準漢字 3384 字

グラフィック画面を構成する「グラフィック V-RAM」は、X1 で 48K バイト、turbo で 96K バイト

イトあります。内訳は、16K バイト(640ドット×200ドット 1色に相当)の RAM が BLUE, RED, GREEN の 3 枚で 48K バイト、さらに turbo にはバンクによって同じ構成のものがもう 1 組で計 96K バイトです。グラフィック画面上の 1 ドットは、各 V-RAM の 1 ビットに対応し、BLUE, RED, GREEN, 計 3 ビットの組合せによってドットごとに 8 色の表示が可能です。また、turbo の場合、2 つのバンクを同時に使用することにより、640×400ドット・8 色の表示も可能になっています。

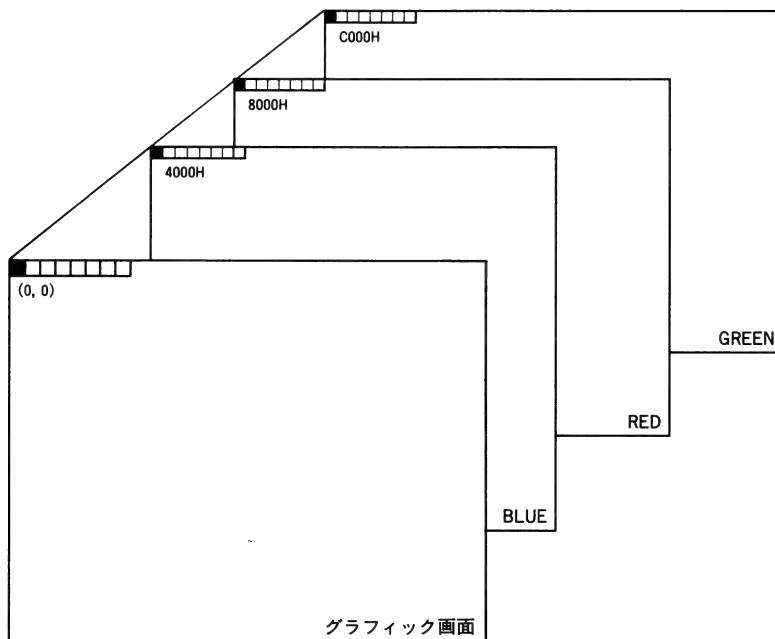


図4-2 グラフィックV-RAMとグラフィック画面概略図

X1 シリーズでは、これらの V-RAM はすべて I/O アドレス上に展開されており、CPU の IN, OUT 命令で読み込み／書き込みを行います。

4-1-1 テキスト V-RAM

テキスト V-RAM は、I/O アドレスの 3000H 番地から 37FFH 番地までで、CG, PCG のフォントを表示する場合には 1 バイトが画面上の 1 文字に対応し、これに書き込まれたデータによって CG または PCG の 256 種類のフォントのなかから一つを選択します。また、漢字を表示する時は 1 バイトが漢字の左、又は右半分に对应し、漢字 ROM アドレスの下位 8 ビットを指示します。

リスト4-1 テキスト V-RAM へのデータ書き込み例

ADRC A2	EQU	18BCH	
WTVRM:	LD	BC, 1A01H	} BIOS ROMバンクがアクティブかどうか?
	IN	A, (C)	
	AND	10H	
	JR	NZ, WTVR1	
	LD	A, 1DHBIOS ROMアクティブ
	JR	WTVR2	
WTVR1:	LD	A, 1EHBIOS ROMノンアクティブ

```

WTVR2:  PUSH      AF
        LD        A, 1DH
        OUT      (00H), A ] BIOS ROMをアクティブにする
        LD        HL, WTVDT
        LD        E, (HL)
        INC      HL
        LD        D, (HL)
        EX       DE, HL
        CALL     ADRCA2 .....カーソル位置のVRAM上のアドレスを得る
        LD        B, H
        LD        C, L
        INC      DE
        LD        A, (DE)
        OUT      (C), A .....データを書き込む
        POP      AF
        OUT      (00H), A ] BIOS ROMの状態を元に戻す
        RET
;
WTVDT:  DB        20H, 10H, 00H
        ;          x座標   y座標   データ
        END

```

リスト4-2 テキストRAMからのデータ読み出し例

```

ADRCA2 EQU      18BCH
RTVRM:  LD        BC, 1A01H
        IN        A, (C) ] BIOS ROMがアクティブor ノンアクティブ
        AND      10H
        JR        NZ, RTVR1
        LD        A, 1DH
        JR        RTVR2 ]
RTVR1:  LD        A, 1EH .....ノンアクティブ
RTVR2:  PUSH     AF
        LD        A, 1DH
        OUT      (00H), A ] ROMをアクティブに
        LD        HL, RTVDT
        LD        E, (HL)
        INC      HL
        LD        D, (HL)
        EX       DE, HL
        CALL     ADRCA2 .....VRAMアドレスを得る
        LD        B, H
        LD        C, L
        IN        A, (C) .....データを読み込む
        INC      DE
        LD        (DE), A
        POP      AF
        OUT      (00H), A ] BIOS ROM選択の状態を元に戻す
        RET
;
RTVDT:  DB        20H, 10H
        DS        1
        ;
        END

```

4-1-2 漢字テキスト V-RAM

X1で漢字を表示するには、オプションの漢字ROMボードを利用してグラフィック画面にパターンとして表示させる必要がありますが、X1turboの場合、漢字ROMをCGやPCGと同列におき、漢字をテキストとして扱うことによって、漢字処理を大幅に簡略化しています。

漢字をテキスト画面に表示するためには、漢字コードに相当する漢字 ROM アドレスをテキスト V-RAM 中で指定してやらなければなりません。漢字は第1水準、第2水準あわせて6802文字ありますから、V-RAM は1文字当り、少なくとも13ビット必要となります。そこで、X1turbo では新しく漢字用テキスト V-RAM2K バイトを I/O アドレスの 3800H 番地から 3FFFH 番地に設け、これを従来のテキスト V-RAM と共に使用することで漢字の表示をおこなっています。



図4-3 漢字用テキストV-RAM(2Kバイト) : I/Oアドレス 3800H~3FFFH

・ D0~D3...ASCII2(漢字 ROM アドレス 2)

漢字 ROM アドレス上位ビットがはいります。(CG, PCG アクセスの場合は無視されます)

・ D4.....1 / 2 水準(漢字 第1 / 第2水準切り換え信号)

(a) 漢字 ROM 選択の場合：第1水準、第2水準のセレクト信号になります。

0 : 第1水準

1 : 第2水準

(b) PCG 選択の場合：CG / KANJI 信号, ROM / RAM 信号とともに、PCG のアクセス方式を選択します。

0 : PCG キャラクタ・モード

1 : PCG 外字モード

(c) CG 選択の場合：無視されます。

・ D5.....ULINE(アンダーライン表示 ON / OFF 信号)

0 : アンダーラインを表示しない

1 : アンダーラインを表示する

・ D6.....LEFT / RIGHT(漢字 左 / 右フォント選択信号)

漢字フォントの左 8 × 16 ドット, 右 8 × 16 ドットのどちらかをアクセスするのセレクトする信号。

0 : LEFT(左半分)

1 : RIGHT(右半分)

CG, PCG アクセスの場合は無視されます。

・ D7.....CG / KANJI(CG / 漢字 ROM 選択信号)

CG と漢字 ROM のどちらをアクセスするのセレクト信号

0 : CG, PCG セレクト

1 : 漢字 ROM, PCG セレクト

4-1-3 アトリビュート V-RAM

アトリビュート V-RAM は、I/O アドレスの 2000H 番地から 27FFH 番地にマッピングされ、表示色や点滅などの文字属性を一字ごとに指定することができます。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
H 倍	V 倍	ROM / RAM	BLINK	REV	G	R	B

• D₀～D₂……キャラクタカラー(8色)を指定します。

D ₂	D ₁	D ₀	ビット 指定色
0	0	0	黒
0	0	1	青
0	1	0	赤
0	1	1	マゼンタ
1	0	0	緑
1	0	1	シアン
1	1	0	黄
1	1	1	白

図4-4 アトリビュートV-RAM(2Kバイト)：I/Oアドレス 2000H～27FFH

• D₀～D₂……COLOR(表示色)

表示色を指定する

• D₃……REV(反転表示信号)

D₀～D₂ で指定したキャラクタを反転表示するための信号

0：指定色のまま表示

1：反転表示

• D₄……BLINK(点滅表示信号)

点滅表示を指定する信号

0：通常表示

1：点滅表示

• D₅……ROM / RAM(ROM / RAM 選択信号)

ROM(CG, 漢字 ROM)と RAM(PCG)のどちらかを表示するかのセレクト信号

0：ROM(CG, 漢字 ROM)表示

1：RAM(PCG)表示

• D₆……V 倍(縦倍表示信号)

キャラクタを縦2倍表示するための信号

0：ノーマル表示

1：縦倍角表示

• D₇……H 倍(横倍表示信号)

キャラクタを横2倍表示するための信号

0：ノーマル表示

1：横倍角表示

次に、テキスト画面に文字を表示させるプログラム例を示します。

リスト4-3 文字表示プログラム

```

ACCDID EQU      179DH
COLORF EQU      0F8D0H
CURSRX EQU      0FADFH
CURSRY EQU      0FAE0H
CRDSP:  LD       BC, 1A01H
        IN       A, (C)
        AND      10H
        JR       NZ, CRDP1
        LD       A, 1DH
        JR       CRDP2
CRDP1:  LD       A, 1EH
CRDP2:  PUSH     AF
        LD       A, 1DH
        OUT      (00H), A
        LD       BC, CDPDT
        LD       A, (BC)
        LD       (CURSRX), A
        INC      BC
        LD       A, (BC)
        LD       (CURSRY), A
        INC      BC
        LD       A, (BC)
        LD       (COLORF), A
        INC      BC
        LD       A, (BC)
        CALL     ACCDIS
        POP      AF
        OUT      (00H), A
        RET
;
CDPDT:  DB       20H, 10H, 04H, 34H
;
        END

```

BIOS ROMの状態を調べ、その状態をセーブ

ROMアクティブ

カーソル位置の設定

色の設定

1文字表示

ROMを元に戻す

x座標 y座標 カラーコード 文字コード

4-1-4 グラフィック V-RAM(X1, X1turbo)

X1のグラフィック V-RAMは、16K バイトずつ BLUE(I/O アドレス 4000H~7FFFH), RED(8000H~BFFFH), GREEN(C000H~FFFFH)の3つの部分に分けられます。画面に表示する時には3つの画面を合成して1つの画面をつくっています。グラフィック画面のX方向8ドットが各 V-RAM の1バイトに、1ドットが1ビットに対応しているので、1ドットごとに3ビット=8色の表示ができます。

X1turbo シリーズは、これとまったく同じ構成のグラフィック V-RAM をバンクによってもう1組もっているので、640×200ドットのグラフィック画面を2画面使うことができます。また、この2組のグラフィック V-RAM を同時に使うことにより、640×400ドット8色の表示が可能です。

バンク切り換えは、画面管理用 I/O ポートに値を設定することによりおこないます。画面管理用 I/O ポートの内容は表 4-1 のようになっていますが、このうちバンク切り換えに関係するのは、ビット 3 (DISP Bank 0/1) とビット 4 (CPU Bank 0/1) の2つです。

DB0: $\overline{\text{L}}/\text{H}$ Res	0: 低解像度モード (200ライン表示) 1: 高解像度モード (400ライン表示)
DB1: $\overline{\text{I}}/2$ RA	0: グラフィック表示=1RA/dot 1: グラフィック表示=2RA/dot(2度打ち)
DB2: $\overline{25}/12$ 行	0: テキスト表示=25行モード (or 20行) 1: テキスト表示=12行モード (or 10行)
DB3: DISP Bank $\overline{0}/1$	0: グラフィックV-RAM Bank 0を画面表示 1: グラフィックV-RAM Bank 1を画面表示
DB4: CPU Bank $\overline{0}/1$	0: グラフィックV-RAM Bank 0をCPUアクセス 1: グラフィックV-RAM Bank 1をCPUアクセス
DB5: SPCG/FPCG	0: コンパチCGアクセスモード (X1とのコンパチモード) 1: 高速CGアクセスモード
DB6: CGSEL $\overline{8}/16\text{RA}$	0: 8 ラスタ/キャラクタのCGファントをCPUアクセス 1: 16ラスタ/キャラクタのCGファントをCPUアクセス
DB7: $\overline{25(12)}/20(10)$ 行	0: テキスト表示=25行 or 12行 1: テキスト表示=20行 or 10行 (アンダーライン表示モード)

表4-1 画面管理用I/Oポート (I/Oアドレス 1FD*H)の内容

・ビット3……DISP Bank(画面表示バンク選択)

0: 表示用グラフィック V-RAM として、バンク 0 を選択する

1: 表示用グラフィック V-RAM として、バンク 1 を選択する

画面に表示するグラフィック V-RAM のバンクを選択する信号です。この値を切り換えることで、画面に表示されているグラフィック画面を瞬時に切り換えることができます。ただし、高解像度(400ライン)モードの場合には、表示用に両方のバンクを使用しているので、この信号は無効になります。

・ビット4……CPU Bank 0/1(CPU アクセスバンク選択)

0: バンク 0 を、CPU からのアクセス可能にする

1: バンク 1 を、CPU からのアクセス可能にする

CPU の読み込み/書き込みを、どちらのバンクに対しておこなうかを指定する信号です。CPU のアクセスするバンクは、表示されているバンクとは関係なく、この信号によって選ぶことができます。

バンクを切り換える場合には、この2つのビットを操作すれば良いのですが、画面管理用I/Oポートにある他のビットを変化させずにポートを書き換えるには、もとの値を知る必要があります。しかし、このポートは書き込み専用ポートなので、直接読み出すわけにはいきません。そこで、XlturboのBIOS ROM ルーチン等では、メインメモリのアドレス F8D6H 番地をバッファとして画面管理用I/Oポートに書き込んだ値を保存しています。したがって、このアドレスを参照することで、画面管理用I/Oポートの現在の値を知ることができます。この点にも注意したバンク切り換えのプログラム例を以下に表示します。

リスト4-4 グラフィックV-RAMバンク切り換えプログラム例

SCRNIO EQU	1FD0H	
WK1FD0 EQU	0F8D6H	
CNGBK: LD	BC, SCRNIO	
LD	A, (WK1FD0)	} アクセスするバンク(ビット4)とディスプレイされて いるバンク(ビット3)をバンク0に
AND	0E7H	
LD	E, A	
XOR	A	


```
LD      HL, CGBDT
LD      D, (HL)
SRA     D
JR      NC, CGBK1
LD      A, 01H
CGBK1:  OR      E
LD      (WK1FD0), A ..... 画面管理情報のセーブ
OUT     (C), A
RET
;
CGBDT:  DB      01H
;
END
```

CGBDTの内容により200ライン, 400ラインの選択

ビット0が立っていれば400ライン, ビット0が立っていなければ200ライン

次に、CPU からグラフィック V-RAM にアクセスする方法について説明します。

X1 シリーズには、CPU からグラフィック V-RAM をアクセスするのに、通常のシングルアクセスモードと turbo シリーズに設定されている同時アクセスモードの 2 つのモードを用意しています。シングルアクセスモードは、一般的に使われている方法で、BLUE, RED, GREEN の 3 つの V-RAM を別々にアクセスするモードです。一方、同時アクセスモードはハードウェア的に I/O アドレス構成を変化させ、3 つの V-RAM の内の 2 つまたは 3 つの V-RAM に対して同時にアクセスしようとする、書き込み専用のモードです。

それぞれのモードの、I/O アドレスマップを図 4-5 に示します。

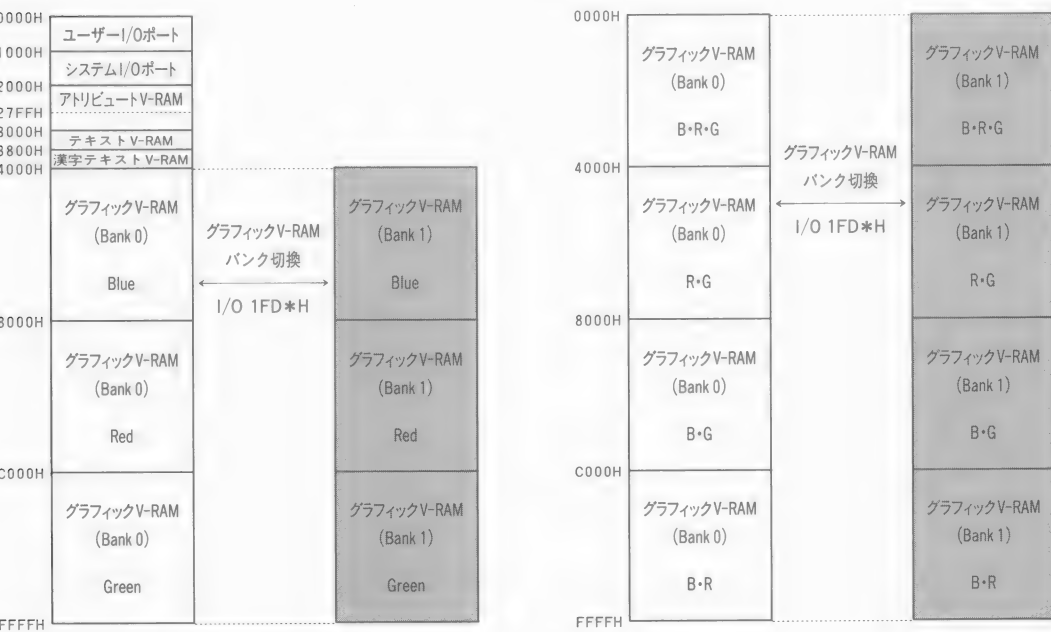


図4-5 シングルアクセスモード及び同時アクセスモードでのI/Oアドレスマップ

2つのアクセスモードを切り換えるには、8255②ポートC・ビット5を使います。

8255②ポートCの内容は、次の表のようになっています。このうち、ビット5のアクセスモード切り換え信号を1から0にすると(正確には、8255②から出ている信号の立ち下がリエッジで)、同時アクセスモードになります。同時アクセスモードからシングルアクセスモードへの復帰は、I/Oポート(どこでもよい)に対する読み込みによっておこなわれます。同時アクセスモードでは、もともと8255②などがあったI/OアドレスにもグラフィックV-RAMを割り当てるため、8255②等に値を設定してモードを切り換えることはできません。そこで、同時アクセスモードが書き込み専用であることを利用して、I/Oに対する読み込みが行われた場合には、シングルアクセスモードへの切り換えと判断するわけです。

ポート	ポート端子	コントロール内容	信号名
C (出力)	PC ₇	立ち上がりでプリンターは入力データをサンプルします。	STROBE
	PC ₆	80/40文字モード (H:40文字モード, L:80文字モード) 基本クロック切り換え	40/80
	PC ₅	I/Oアクセスモード切り換え (同時アクセスモード)	GWRMD
	PC ₄	スムーズスクロール信号	スムーズスクロール
	PC ₃		—
	PC ₂		—
	PC ₁		—
	PC ₀	カセットテープへの書き込みデータ	WRITE DATA

表4-2 (8255②ポートCの内容)

・ビット5……アクセスモード切り換え信号

0：同時アクセスモードに切り換え

同時アクセスモードからシングルアクセスモードに切り換えるには、I/Oアドレス(どこでもよい)に対する読み込み命令を実行します。

グラフィックV-RAMは、すべてI/Oアドレスに展開されています。したがって、グラフィックV-RAMをアクセスする場合にはCPUのIN、OUT命令を使います。

同時アクセスモードの場合でも、アドレス構成が変わる点と、読み込み(IN命令)を実行するとシングルアクセスモードに戻ってしまう点が違うだけで、書き込み方はシングルアクセスモードと同じです。

第4章 画面表示

(1) 25行モード

X座標		0	1	39
Y座標	0	2000 (2000) (3800)	3001 (2001) (3801)	3027 (2027) (3827)
	1	3028 (2028) (3828)	3029 (2029) (3829)	304F (204F) (384F)
		∫	∫	∫
	24	33C0 (23C0) (38C0)	33C1 (23C1) (38C1)	33E7 (23E7) (38E7)

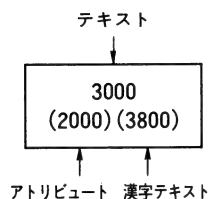
40文字×25行モード(ページ0)

X座標		0	1	39
Y座標	1	3400 (2400) (3C00)	3401 (2401) (3C01)	3427 (2427) (3C27)
	2	3428 (2428) (3C28)	3429 (2429) (3C29)	344F (244F) (3C4F)
		∫	∫	∫
	24	37C0 (27C0) (3FC0)	37C1 (27C1) (3FC1)	37E7 (27E7) (3FE7)

40文字×25行モード(ページ1)

X座標		0	1	79
Y座標	0	3000 (2000) (3800)	3001 (2001) (3801)	304F (204F) (384F)
	1	3050 (2050) (3850)	3051 (2051) (3851)	309F (209F) (389F)
		∫	∫	∫
	24	3780 (2780) (3F80)	3781 (2781) (3F81)	37CF (27CF) (3FCF)

80文字×25行モード



(2) 12行モード

X座標		0	1	39
Y座標	0	3000 (2000) (3800)	3001 (2001) (3801)	3027 (2027) (3827)
	1	3028 (2028) (3828)	3029 (2029) (3829)	304F (204F) (384F)
		∫	∫	∫
	11	31B8 (21B8) (39B8)	31B9 (21B9) (39B9)	31DF (21DF) (39DF)

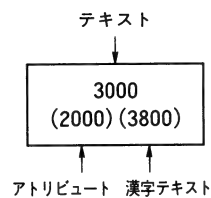
40文字×12行モード(ページ0)

X座標		0	1	39
Y座標	0	3200 (2200) (3A00)	3201 (2201) (3A01)	3227 (2227) (3A27)
	1	3228 (2228) (3A28)	3229 (2229) (3A29)	324F (224F) (3A4F)
		∫	∫	∫
	11	33B8 (23B8) (3BB8)	33B9 (23B9) (3BB9)	33DF (23DF) (3BDF)

40文字×12行モード(ページ1)

X座標		0	1	79
Y座標	0	3000 (2000) (3800)	3001 (2001) (3801)	304F (204F) (384F)
	1	3050 (2050) (3850)	3051 (2051) (3851)	309F (209F) (389F)
		∫	∫	∫
	11	3370 (2370) (3B70)	3371 (2371) (3B71)	33BF (23BF) (3BBF)

80文字×12行モード



(3) 20行モード

X座標		0	1	39
Y座標	0	3000 (2000) (3800)	3001 (2001) (3801)	3027 (2027) (3827)
	1	3028 (2028) (3828)	3029 (2029) (3829)	304F (204F) (384F)
		∫	∫	∫
	19	32F8 (22F8) (3AF8)	32F9 (22F9) (3AF9)	331F (231F) (3B1F)

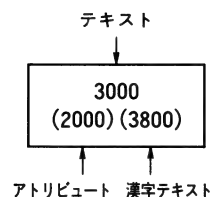
40文字×20行モード(ページ0)

X座標		0	1	39
Y座標	0	3400 (2400) (3C00)	3401 (2401) (3C01)	3427 (2427) (3C27)
	1	3428 (2428) (3C28)	3429 (2429) (3C29)	344F (244F) (3C4F)
		∫	∫	∫
	19	36F8 (26F8) (3EF8)	36F9 (26F9) (3EF9)	371F (271F) (3F1F)

40行×20行モード(ページ1)

X座標		0	1	79
Y座標	0	3000 (2000) (3800)	3001 (2001) (3801)	304F (204F) (384F)
	1	3050 (2050) (3850)	3051 (2051) (3851)	309F (209F) (389F)
		∫	∫	∫
	19	35F0 (25F0) (3DF0)	35F1 (25F1) (3DF1)	363F (263F) (3E3F)

80文字×20行モード



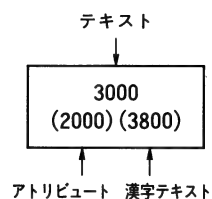
(4) 10行モード

X座標		0	1	39
Y座標	0	3000 (2000) (3800)	3001 (2001) (3801)	3027 (2027) (3827)
	1	3028 (2028) (3828)	3029 (2029) (3829)	304F (204F) (384F)
		∫	∫	∫
	9	3168 (2168) (3968)	3169 (2169) (3969)	318F (218F) (398F)

40文字×10行モード(ページ0)

X座標		0	1	39
Y座標	0	3200 (2200) (3A00)	3201 (2201) (3A01)	3227 (2227) (3A27)
	1	3228 (2228) (3A28)	3229 (2229) (3A29)	324F (224F) (3A4F)
		∫	∫	∫
	9	3368 (2368) (3B68)	3369 (2369) (3B69)	338F (238F) (3B8F)

40文字×10行モード(ページ1)















X座標		0	1	79
Y座標	10	3000 (2000) (3800)	3001 (2001) (3801)	304F (204F) (384F)
	1	3050 (2050) (3850)	3051 (2051) (3851)	309F (209F) (389F)
		∫	∫	∫
	9	32D0 (22D0) (3AD0)	32D1 (22D1) (3AD1)	331F (231F) (3B1F)

80文字×10行モード













図4-6 テキスト系V-RAMアドレスと画面表示位置

第4章 画面表示













(1) 200ドットモード

		1文字表示分(8ドット)		
1行8ラスタ	4000	4001	〜	4027
	4800	4801	〜	4827
	5000	5001	〜	5027
	5800	5801	〜	5827
	6000	6001	〜	6027
	6800	6801	〜	6827
	7000	7001	〜	7027
	7800	7801	〜	7827
	4028	4029	〜	404F
				
	7828	7829	〜	784F
				
	43C0	43C1	〜	43E7
				
	7BC0	7BC1	〜	7BE7

320×200ドット (BLUEページ0)

4400	4401	〜	4427
4C00	4C01	〜	4C27
5400	5401	〜	5427
5C00	5C01	〜	5C27
6400	6401	〜	6427
6C00	6C01	〜	6C27
7400	7401	〜	7427
7C00	7C01	〜	7C27
4428	4429	〜	444F
			
7C28	7C29	〜	7C4F
			
47C0	47C1	〜	47E7
			
7FC0	7FC1	〜	7FE7

320×200ドット (BLUEページ1)

4000	4001	〜	404F
4800	4801	〜	484F
5000	5001	〜	504F
5800	5801	〜	584F
6000	6001	〜	604F
6800	6801	〜	684F
7000	7001	〜	704F
7800	7801	〜	784F
4050	4051	〜	409F
			
7850	7851	〜	789F
			
4780	4781	〜	47CF
			
7F80	7F81	〜	7FCF

640×200ドット (BLUE)

(2) 192ドットモード

1文字表示分(8ドット)			
4000	4001	〜	4027
4400	4401	〜	4427
4800	4801	〜	4827
4C00	4C01	〜	4C27
7800	7801	〜	7827
7C00	7C01	〜	7C27
4028	4029	〜	404F
7C28	7C29	〜	7C4F
41B8	41B9	〜	41DF
7DB8	7DB9	〜	7DDF

320×192ドット (BLUEページ0)

4200	4201	〜	4227
4600	4601	〜	4627
4A00	4A01	〜	4A27
4E00	4E01	〜	4E27
7A00	7A01	〜	7A27
7E00	7E01	〜	7E27
4228	4229	〜	424F
7E28	7E29	〜	7E4F
43B8	43B9	〜	43DF
7FB8	7FB9	〜	7FDF

320×192ドット (BLUEページ1)

4000	4001	〜	404F
4400	4401	〜	444F
4800	4801	〜	484F
4C00	4C01	〜	4C4F
7800	78C1	〜	784F
7C00	7C01	〜	7C4F
4050	4051	〜	409F
7C50	7C51	〜	7C9F
4370	4371	〜	43BF
7F70	7F71	〜	7FBF

640×192ドット (BLUE)

(3) 400ドットモード

1文字表示分(8ドット)			
4000	4001	〜	4027
4000'	4001'	〜	4027'
4800	4801	〜	4827
4800'	4801'	〜	4827'
7800	7801	〜	7827
7800'	7801'	〜	7827'
4028	4029	〜	404F
4028'	4029'	〜	404F'
7828	7829	〜	784F
7828'	7829'	〜	784F'
43C0	43C1	〜	43E7
7BC0	7BC1	〜	7BE7

320×400ドット (BLUEページ0)

4400	4401	〜	4427
4400'	4401'	〜	4427'
4C00	4C01	〜	4C27
4C00'	4C01'	〜	4C27'
7C00	7C01	〜	7C27
7C00'	7C01'	〜	7C27'
4428	4429	〜	444F
4428'	4429'	〜	444F'
7C28	7C29	〜	7C4F
7C28'	7C29'	〜	7C4F'
47C0	47C1	〜	47E7
7FC0	7FC1	〜	7FE7

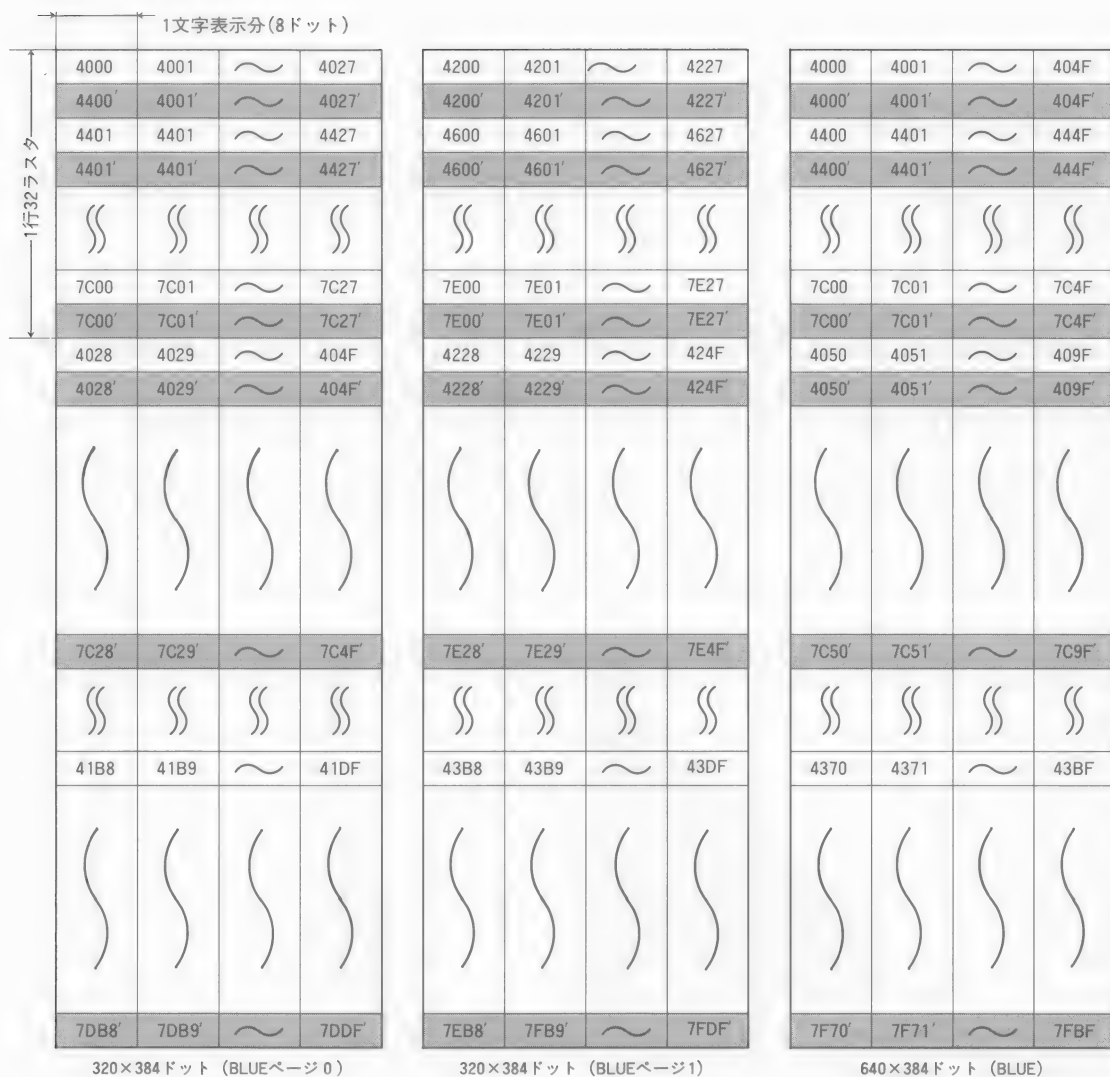
320×400ドット (BLUEページ1)

4000	4001	〜	404F
4000'	4001'	〜	404F'
4800	4801	〜	484F
4800'	4801'	〜	484F'
7800	7801	〜	784F
7800'	7801'	〜	784F'
4050	4051	〜	409F
4050'	4051'	〜	409F'
7850	7851	〜	789F
7850'	7851'	〜	789F'
4780	4781	〜	47CF
7F80	7F81	〜	7FCF

640×400ドット (BLUE)

4000 バンク0
4000' バンク1

(4) 384ドットモード



4000 バンク0

4000' バンク1

図4-7 グラフィックV-RAMアドレスと画面表示位置

4-1-5 グラフィック V-RAM(X1turboZ)

X1turboZ の画面表示モードには、「コンパチモード」と「多色モード」があります。コンパチモードでは、グラフィックはX1turbo と全く同じになります。多色モードでは、V-RAM と画面との対応が変わりますが、V-RAM へのアクセス方法などはコンパチモードと同じです。

(1) 多色表示の原理

8 色表示の場合は、B (青)、R (赤)、G (緑) の三原色がそれぞれ「ある」か「ない」かの 8 通りの組合せで色を表示していました。これに対し多色モードでは、三原色のそれぞれの色の明るさ(階調)を指定することによって多くの色を表示します。

図4-8 は、多色モードの時、B、R、G が各々何階調で表現されているかを示したものです。8 色モードでは各色 2 階調(1 ビット)ですが、64 色モードの時は各色 4 階調(2 ビット)となります。表現できる色の数は、4 階調の 3 乗=64 色です。同様に 4096 色モードでは、各色 16 階調(4 ビット)ですから、表現できる色の数は 16 の 3 乗で 4096 色と計算されます。

8 色を表現するためには、1 ドット当り 3 ビットの情報が必要でしたが、64 色の時は 1 ドット当り 6 ビット、4096 色の時は 1 ドット当り 12 ビットの情報が必要です。多色モードを使用する際は、これらの情報が V-RAM 上にどのように配置されているか把握している必要があります。

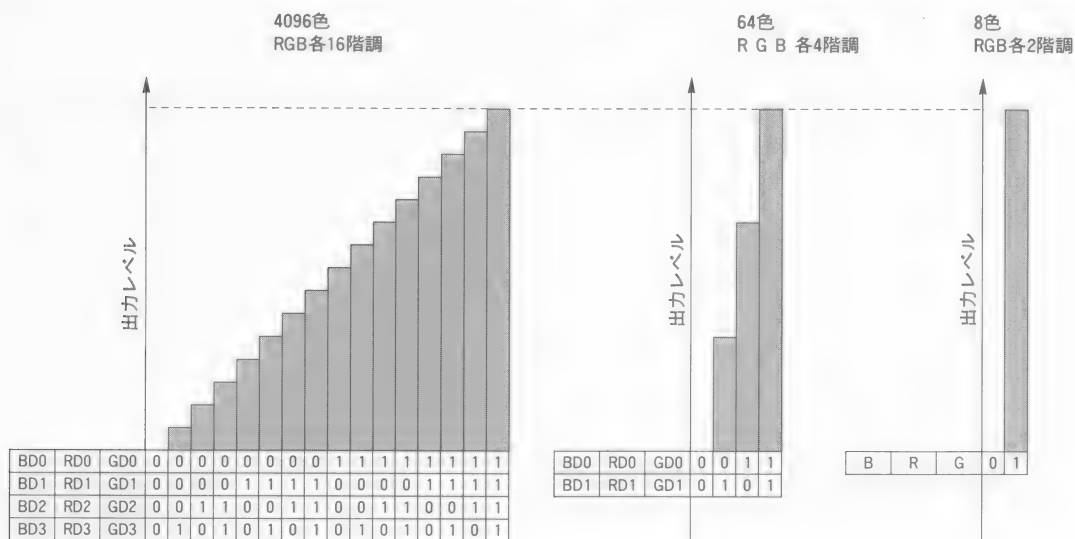


図4-8

(2) グラフィック V-RAM のアドレス領域

多色モード時には、画面上のドットと V-RAM のアドレスとの関係はコンパチモードと較べて大きく変化します。ここでは、多色モード時のグラフィック V-RAM アドレスについて説明します。以下、アドレスはグラフィック V-RAM の BLUE のデータについてのみ示しますので、RED および GREEN のアドレスについては、図のアドレスに対して以下の計算をしてください。

BLUE のアドレス=図のアドレス

RED のアドレス=図のアドレス+4000H

GREENのアドレス=図のアドレス+8000H

(a) 低解像度 40×25 行：320×200 ドット 4096 色 1 画面

4096色モードでのV-RAMアドレスを図4-9に示します。この図からわかるように、各ドットに対してB, R, G合わせて12個のアドレスが対応しています。この時、図4-10に示すようにバンク0のページ0がBD0に、バンク0のページ1がBD1に、バンク1のページ0がBD2に、バンク1のページ1がBD3にそれぞれ対応しています。画面上の表示位置とV-RAMアドレスの関係は次のようになります。

BD0のアドレス=バンク0, ページ0での表示位置

BD1のアドレス=バンク0で、BD0のアドレス+400H

BD2のアドレス=バンク1で、BD0のアドレス

BD3のアドレス=バンク1で、BD0のアドレス+400H

RED, GREENについても、前述の計算をすることでアドレスを求めることができます。

このモードでは、ビデオ画像等の映像入力が可能です。

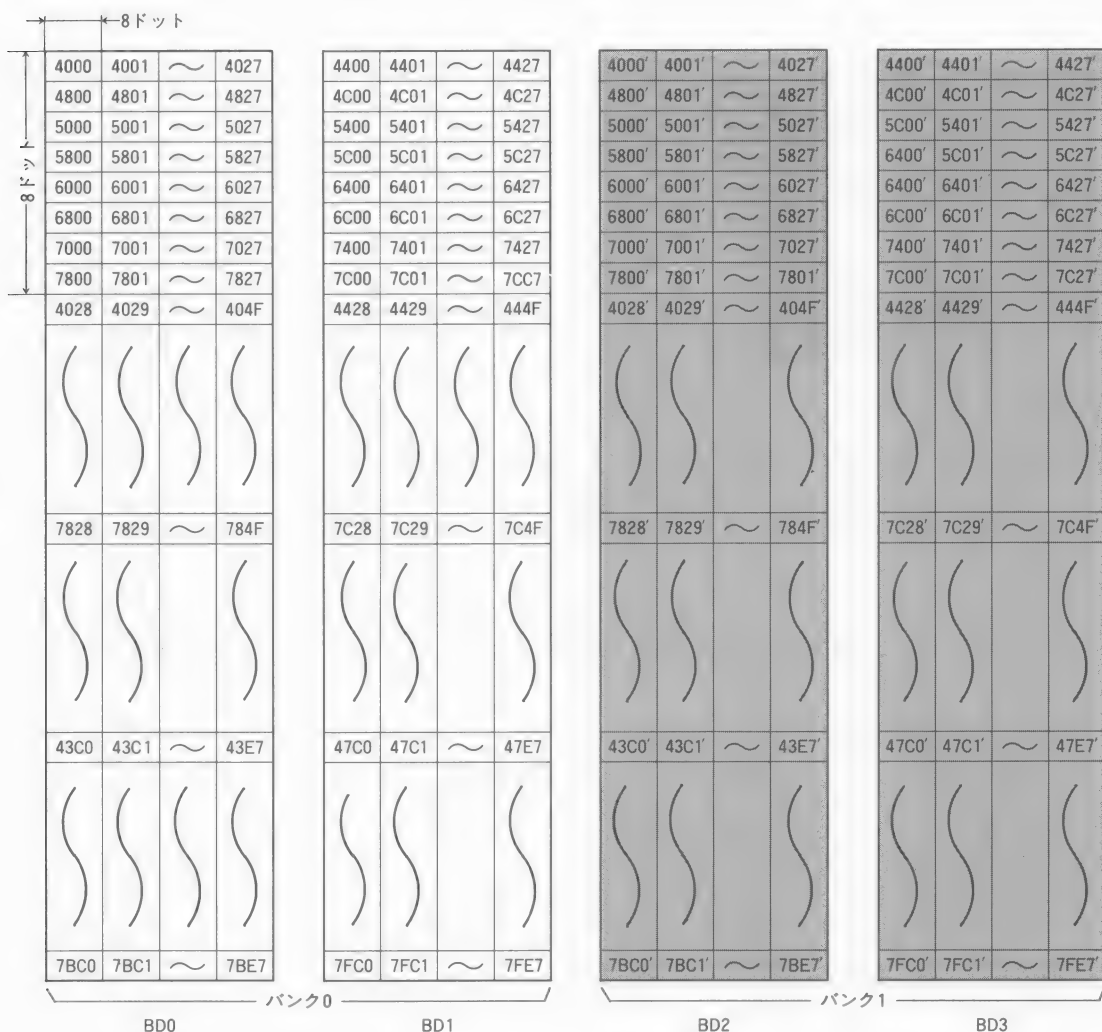


図4-9 低解像度320×200ドット 4096色1画面

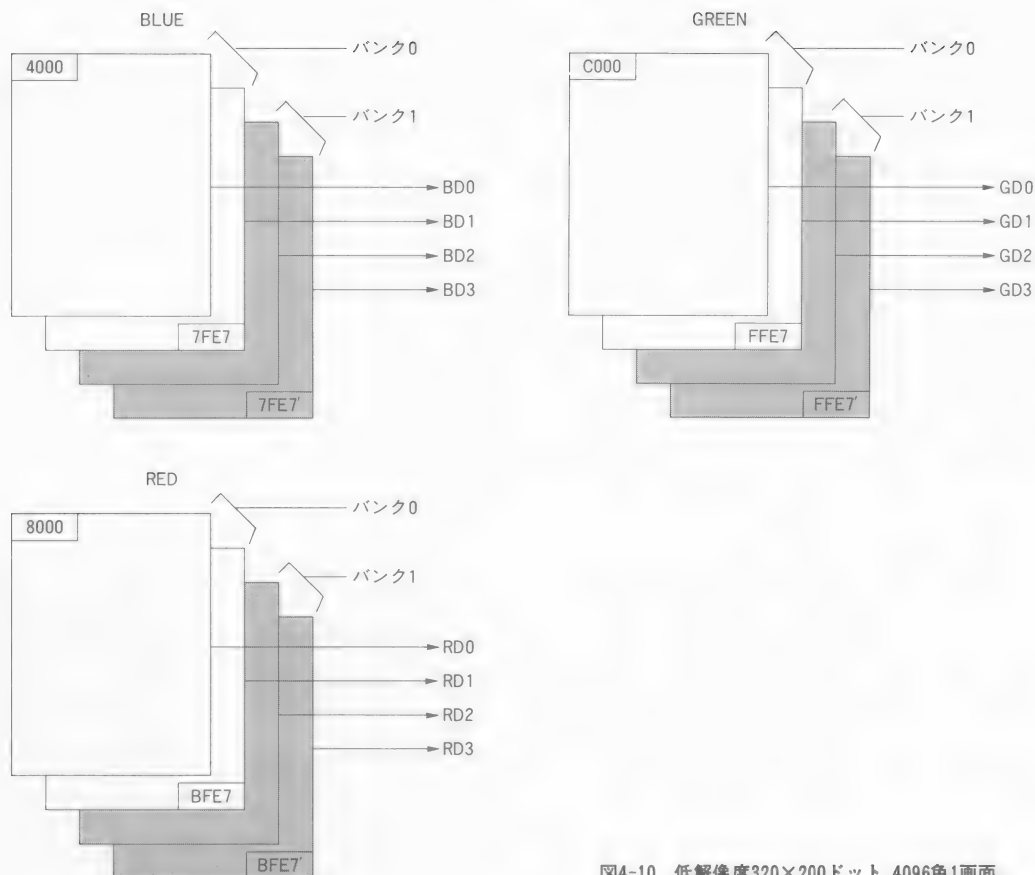


図4-10 低解像度320×200ドット 4096色1画面

(b) 低解像度 40×25行：320×200ドット 64色 2画面

この表示モードでは、64色2画面を、優先順位をつけて同時に表示することが可能です。このモードでは、モードの切り換えを行った後、拡張パレットを設定しなおさないと正しく色が表示されません。

このモードにおけるグラフィック V-RAM のアドレス構成を図4-11に示します。表示される2画面を、それぞれ第1画面、第2画面とすると、表示位置と V-RAM アドレスの関係は次のようになります。

グラフィック第1画面

BD 0 のアドレス=バンク 0， ページ 0 での表示位置

BD 1 のアドレス=バンク 0 で、BD0 のアドレス+400H

グラフィック第2画面

BD 0 のアドレス=バンク 1， ページ 0 での表示位置

BD 1 のアドレス=バンク 1 で、BD0 のアドレス+400H

同様の関係が、それぞれ RED， GREEN に対しても成り立ちます。

このモードでは、ビデオ画像等の映像入力が可能です。

(c) 低解像度 80×25 行：640×200 ドット 64 色 1 画面

この表示モードでは64色1画面を表示することが可能です。このモードにおけるグラフィック V-RAM のアドレス構造を図 4-12 に示します。画面表示位置と V-RAM アドレスとの関係は次の通りです。

BD 0 のアドレス=バンク 0 での表示位置

BD 1 のアドレス=バンク 1 で BD0 のアドレス

この関係は RED, GREEN についても同様に成り立ちます。

このモードでは、ビデオ画像等の映像入力が可能です。

(d) 高解像度 40×25 行：320×200 ドット 64 色 2 画面

このモードにおけるグラフィック V-RAM のアドレス構成を図 4-11 に示します。画面表示位置と V-RAM アドレスとの関係は次の通りです。

グラフィック第1画面

BD 0 のアドレス=バンク 0, ページ 0 での表示位置

BD 1 のアドレス=バンク 0 で, BD0 のアドレス+400H

グラフィック第2画面

BD 0 のアドレス=バンク 1, ページ 0 での表示位置

BD 1 のアドレス=バンク 1 で, BD0 のアドレス+400H

同様の関係が、第1画面、第2画面の両方において、それぞれ RED, GREEN に対しても成り立ちます。

また、このモードにおいて、2画面同時表示を指定しても無効となります。

(e) 高解像度 80×25 行：640×200 ドット 64 色 1 画面

この表示モードは、64色1画面を表示するモードです。このモードにおけるグラフィック V-RAM のアドレスの構成を図 4-12 に示します。画面表示位置と V-RAM アドレスとの関係は次の通りです。

BD 0 のアドレス=バンク 0 での表示位置

BD 1 のアドレス=バンク 1 で, BD0 のアドレス

この関係は、RED, GREEN についても同様に成り立ちます。

また、このモードではビデオ画像等の映像入力およびスーパーインポーズはできません。

(f) 高解像度 40×15 行：320×400 ドット 64 色 1 画面

この表示モードは、64色1画面を表示するモードです。このモードにおけるグラフィック V-RAM とアドレスの関係を図 4-13 に示します。画面表示位置と V-RAM との関係は次の通りです。

BD 0 のアドレス=ページ 0 での表示位置

BD 1 のアドレス=BD0 のアドレス+400H

この関係は、RED, GREEN についても同様に成り立ちます。

このモードでは1ラスタごとに V-RAM のバンクを切り換えて表示しています。また、このモードではビデオ画像等の映像入力およびスーパーインポーズはできません。

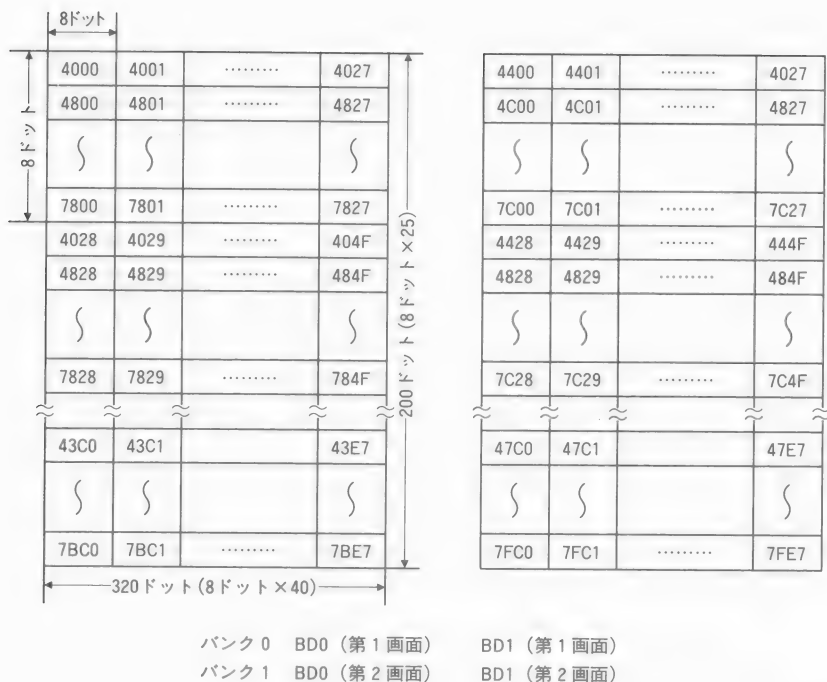


図4-11 低/高解像度 320×200ドット 64色 2画面

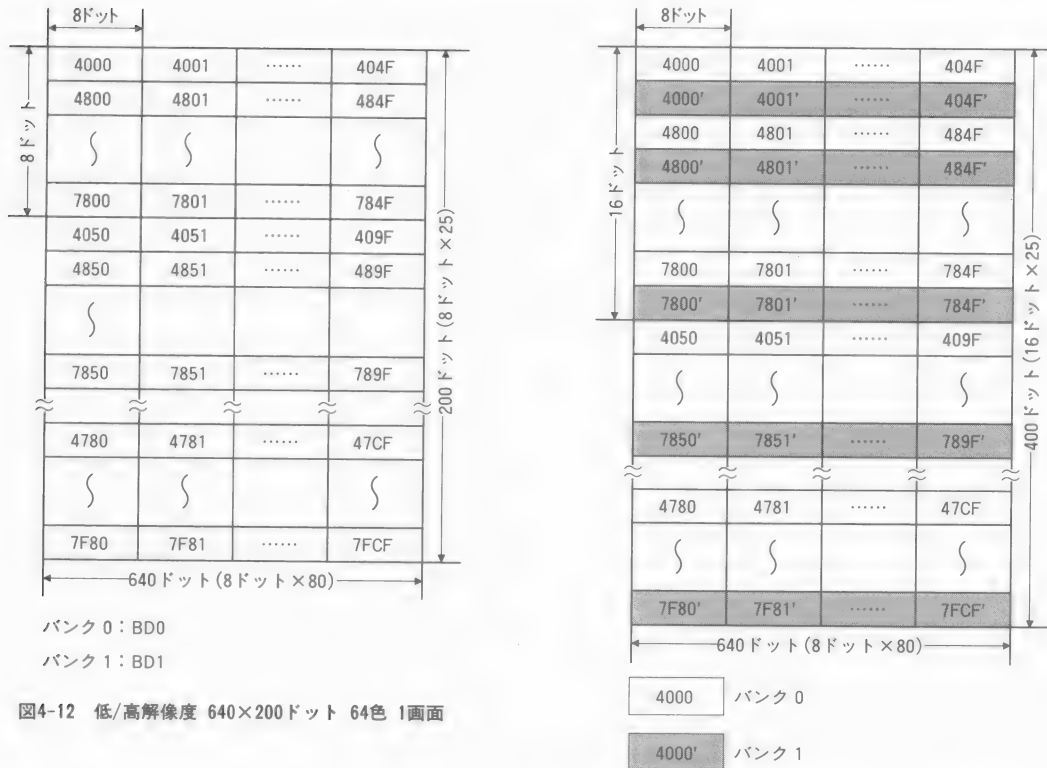


図4-12 低/高解像度 640×200ドット 64色 1画面

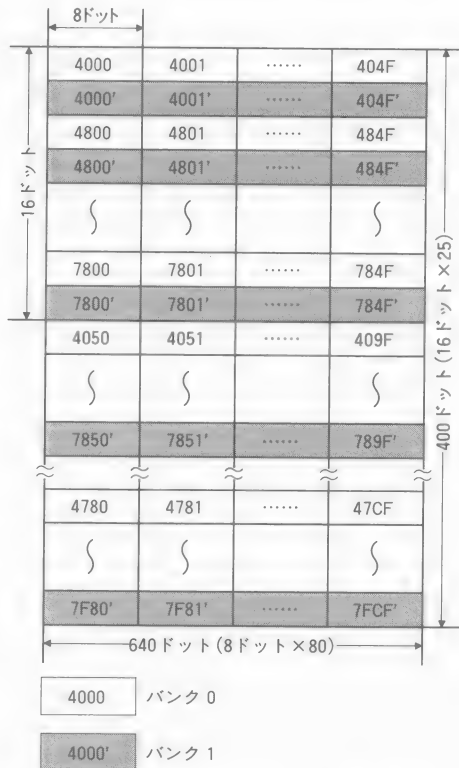


図4-14 高解像度 640×400ドット 8色 1画面

(g) 高解像度 80×25 行 : 640×400 8 色 1 画面

このモードにおけるグラフィック V-RAM のアドレス構成を図 4-14 に示します。

このモードでは 1 ラスタごとに V-RAM のバンクを切り換えて表示しています。このモードは X1turbo の 640×400 の画面と同一のアドレス構成になっていますが、拡張パレットの設定を行うことができるようになっています。

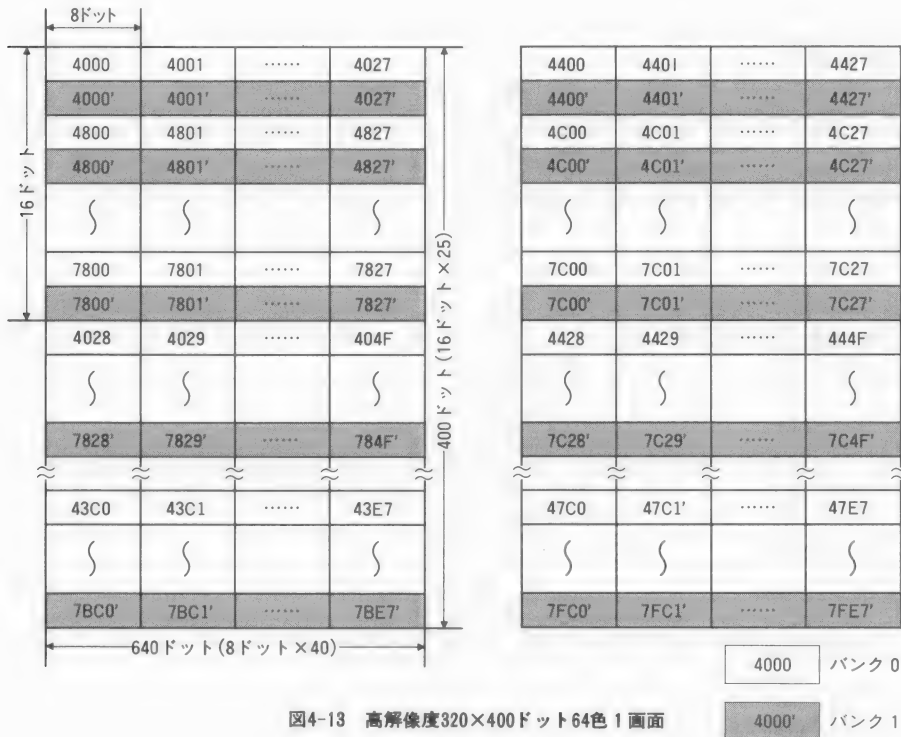


図4-13 高解像度320×400ドット64色1画面

4-2 画面の構成

4-2-1 CRT コントローラ概要

X1 シリーズでは、画面表示制御に CRT コントローラ (CRTC) HD46505-SP を使用しています。

CRTC は、ディスプレイ用の同期信号の生成や V-RAM のリフレッシュアドレスの生成、表示タイミングの制御、カーソル制御、ライトペン制御などの機能を持っています。X1 シリーズでは、これらの機能のうち、

- (1) 水平・垂直同期信号の生成、表示タイミングの制御
- (2) 表示画面の大きさ
- (3) 画面および V-RAM のリフレッシュ

など画面制御に関する機能を使用しています。制御内容のおもな仕様は以下のとおりです。

項 目	低解像度モード	高解像度モード
走 査 方 式	ノンインターレース・モード	
水平同期周波数	15.98KHz	24.86KHz
水平同期信号幅	4.47μsec	2.98μsec
垂直同期周波数	61.9Hz	55.5Hz
垂直同期信号幅	188μsec	321.8μsec

表4-3 画面の仕様

4-2-2 CRTC のコントロール

HD46505-SP は、各種の制御値を保持しておくための18個の内部レジスタ(R0～R17)と、それを選択するためのアドレスレジスタ(AR)で構成されています。このうち R0～R9 は、同期、表示画面の構成、表示のタイミング等の設定に使用され、R10～R17 は、カーソル表示制御、ライトペン制御、表示開始アドレスの設定等に使用されます。

以下に CRT コントローラのレジスタと CRT 画面構成を示します。

レジスタ番号	レジスタ名称	機 能	書き込み値																																																	
R0	水平総文字数	水平走査の周期を指定します。	Nht *																																																	
R1	水平表示文字数	1 行当りの表示文字数を指定します。	Nhd																																																	
R2	水平同期位置	水平同期信号の出力位置を指定します。 水平同期位置をH文字目にするとき、(H-1)を設定します。	Nhsp *																																																	
R3	同期パルス幅	下位 4 ビットで水平同期信号のパルス幅（基本単位：水平 1 文字時間CH）を、上位 4 ビットで垂直同期信号のパルス幅（基本単位：水平 1 走査時間H）を指定します。	Nvsw Nhsw																																																	
R4	垂直総文字数	垂直走査の文字数を指定します。	Nvt *																																																	
R5	総ラスタ調整	1 フレーム当りの総ラスタ数を調整するため、1 フィールドの最後に付加するラスタ数を指定します。	Nadj																																																	
R6	垂直表示文字数	画面に表示する文字行数を指定します。 垂直総文字数より小さい数値を設定します。	Nvd																																																	
R7	垂直同期位置	垂直同期信号の出力位置を指定します。	Nvsp *																																																	
R8	インターレース&スキュー (遅れ)	<div>ラスタスキャンモード指定と、CUDISP信号、DISPTMG信号のスキューを指定します。</div> <div>ビット 7 6 5 4 3 2 1 0</div> <table><tr><td>C₁</td><td>C₀</td><td>D₁</td><td>D₀</td><td>*</td><td>*</td><td>V</td><td>S</td></tr></table> <div>*印無効ビット</div> <div>インターレースモード</div> <table><tr><td>V</td><td>S</td><td>ラスタスキャンモード</td></tr><tr><td>0</td><td>0</td><td rowspan="2">} ノンインターレースモード</td></tr><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>インターレースシンクモード</td></tr><tr><td>1</td><td>1</td><td>インターレースシンク&ビデオモード</td></tr></table> <div>DISPTMGスキュービット</div> <table><tr><td>D₁</td><td>D₀</td><td>DISPTMG信号</td></tr><tr><td>0</td><td>0</td><td>スキューなし</td></tr><tr><td>0</td><td>1</td><td>1 文字スキュー</td></tr><tr><td>1</td><td>0</td><td>2 文字スキュー</td></tr><tr><td>1</td><td>1</td><td>出力されない</td></tr></table> <div>C₁ C₀ CUDISP信号</div> <table><tr><td>0</td><td>0</td><td>スキューなし</td></tr><tr><td>0</td><td>1</td><td>1 文字スキュー</td></tr><tr><td>1</td><td>0</td><td>2 文字スキュー</td></tr><tr><td>1</td><td>1</td><td>出力されない</td></tr></table>	C ₁	C ₀	D ₁	D ₀	*	*	V	S	V	S	ラスタスキャンモード	0	0	} ノンインターレースモード	1	0	0	1	インターレースシンクモード	1	1	インターレースシンク&ビデオモード	D ₁	D ₀	DISPTMG信号	0	0	スキューなし	0	1	1 文字スキュー	1	0	2 文字スキュー	1	1	出力されない	0	0	スキューなし	0	1	1 文字スキュー	1	0	2 文字スキュー	1	1	出力されない	
C ₁	C ₀	D ₁	D ₀	*	*	V	S																																													
V	S	ラスタスキャンモード																																																		
0	0	} ノンインターレースモード																																																		
1	0																																																			
0	1	インターレースシンクモード																																																		
1	1	インターレースシンク&ビデオモード																																																		
D ₁	D ₀	DISPTMG信号																																																		
0	0	スキューなし																																																		
0	1	1 文字スキュー																																																		
1	0	2 文字スキュー																																																		
1	1	出力されない																																																		
0	0	スキューなし																																																		
0	1	1 文字スキュー																																																		
1	0	2 文字スキュー																																																		
1	1	出力されない																																																		

R9	最大ラスタアドレス	行間のスペースを含めた 1 行のラスタ数を指定します。	Nr
R10 R11	カーソルスタートラスタ カーソルエンドラスタ	この 2 つのレジスタにより、カーソルの形状および表示モードを指定します。X1 では、このカーソル制御機能を使用していません。	
R12 R13	スタートアドレス	リフレッシュメモリの読み出し先頭アドレスを指定します。	
R14 R15	カーソルアドレス	カーソルの表示アドレスを指定します。 X1 では使用していません。	
R16 R17	ライトペン	ライトペンの検出アドレスを記憶するレジスタです。 X1 では使用していません。	

* のついたレジスタは {書きこみ値} = {指定値} - 1

表4-4 CRTコントローラの内部レジスタ

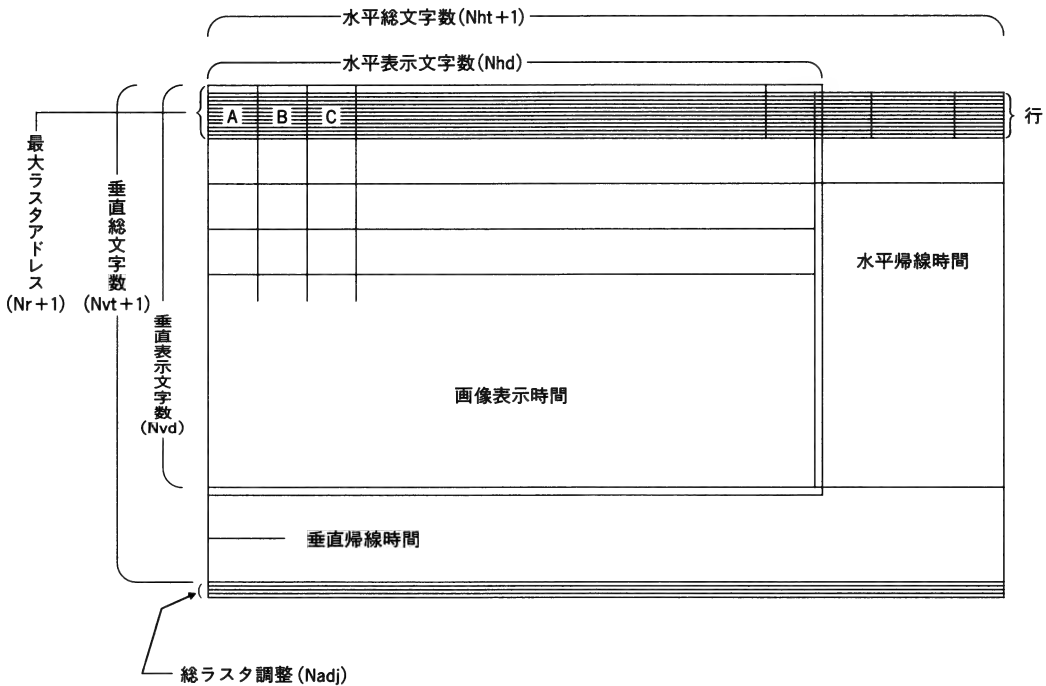


図4-15 CRTCの内部レジスタとCRT画面構成の関係

CRTC の内部レジスタに値を設定するには、まずアドレスレジスタ (AR) に設定しようとする内部レジスタの番号を書き込み、それからレジスタデータを書き込みます。アドレスレジスタはシステム I/O ポートの 1800H 番地に、レジスタデータは同じく 1801H 番地にそれぞれ割り振られています。次に CRTC のアクセス例を示します。

リスト4-5 CRTCアクセス例

```
CRTCAD EQU 1800H .....CRTCレジスタ指定ポートI/Oアドレス
CRCAC: LD BC, CRTCAD
      LD HL, CRCDT
      LD A, (HL)
      OUT (C), A
      INC HL
      INC BC
      LD A, (HL)
      OUT (C), A
      RET
;
CRCDT: DB 0FH, 00H
      ;
      END
```

書き込むデータ
CRTCレジスタ番号

また、CRTCはRESETしても、内部レジスタの値はクリアされないので、電源投入時には必ずイニシャライズしなければなりません。次にCRTCのイニシャライズ例を示します。

リスト4-6 CRTCイニシャライズ例

```
CRTCAD EQU 1800H
CRCIN: LD BC, CRTCAD
      LD HL, CRIDT
      XOR A
      OUT (C), A .....CRTC-Reg-No. 指定
      INC BC
      LD D, (HL)
      OUT (C), D .....データを書き込む
      INC A
      INC HL
      DEC BC
      CP 10H
      JR NZ, CRCI1
      RET
;
CRIDT: DB
      DB
      ;
      END
```

40字モード 表示文字数 同期パルス幅 トータルラスタアジャスト 垂直文字数 垂直表示文字数 垂直同期位置

37H	28H	2DH	34H	1FH	02H	19H	1CH
00H	07H	60H	07H	00H	00H	00H	00H

インターレース 最大ラスタアドレス カーソルスタートラスタ カーソルエンドラスタ スタートアドレス(H) カーソル(H) カーソル(L)

CRTCのRegisterに書き込むデータを並べている

4-2-3 表示画面モードの切り換え

X1は次の表に示すような画面構成になっています。

テキスト画面	グラフィック画面	面数
40×25	320×200	2
80×25	640×200	1

表4-5 X1の画面構成

テキスト画面を40文字×25行に設定すると、グラフィック画面は自動的に320×200ドットとなり、テキスト画面を80文字×25行に設定するとグラフィック画面は640×200ドットとなります。色の指定はどちらのモードでも、テキスト画面は文字単位に、またグラフィック画面はドット単位に8色の指定ができます。

X1turbo シリーズは次の表に示すような画面構成になっています

(a) 低解像度の表示画面モード (200ラインモニター用)

テキスト画面	コンパチモード画面 (X1turbo)	色数	面数	多色モード画面 (X1turboZ)	色数	面数
40×25	320×200	8	4	320×200	4096	1
40×25	320×200	8	4	320×200	64	2
40×12	320×192	8	4	※注		
40×20	-----			-----		
40×10	-----			-----		
80×25	640×200	8	2	640×200	64	1
80×12	640×192	8	2	※注		
80×20	-----			-----		
80×10	-----			-----		

(b) 高解像度の表示画面モード (400ラインモニター用)

テキスト画面	コンパチモード画面 (X1turbo)	色数	面数	多色モード画面 (X1turboZ)	色数	面数
40×25	320×200	8	4	320×200	64	2
40×12	320×192	8	4	※注		
40×20	-----			-----		
80×25	640×200	8	2	640×200	64	1
80×12	640×192	8	1	※注		
80×20	-----			-----		
40×25	320×400	8	2	320×400	64	1
40×12	320×384	8	2	※注		
80×25	640×400	8	1	640×400		1
80×12	640×384	8	1	※注		

※注 グラフィックは、縦方向に2ドットずつ表示されます。特に必要でない限り使用しないで下さい。

表4-6: X1turboシリーズの画面構成

X1turbo と X1turboZ では、テキスト画面構成は全く同じです。また X1turboZ のコンパチモードでは、グラフィック画面構成は X1turbo と共通になっています。

「多色モード」は X1turboZ 独自のモードですが、データの出力が異なっているだけで、画面構成はコンパチモードと全く同じになっています。この時、バンクおよびページに分けられたB、R、Gの各データを同時に出力することにより多色表現を可能としています。また、低解像度の320×200ドットにおいて、モード指定ポート(1FB0H)のD3(2P)が1のとき、64色2画面としてテキスト画面とこの2つのグラフィック画面との間で優先順位をきめて同時に表示することが可能です。詳しくはプライオリティの項を参照してください。

(1) 40(80)×25 行：320(640)×200 ドット 表示モード

高解像度・低解像度モード、両方にあるモードです。

この表示モードでは、従来の X1 の表示モードと V-RAM のアドレス構成が同じになります。したがって、従来の X1 のソフトウェアは、このモードで実行することができます。ただし、フルコンパチになるのは低解像度モードのときだけで、高解像度モードの場合には、特別な操作が必要になることがあります。低解像度モードのときは、漢字を表示することができませんが、高解像度モードでは正常に表示されます。アンダーラインの表示はできません。

(2) 40(80)×12 行：320(640)×192 ドット 表示モード

高解像度・低解像度モードの両方にあるモードです。

この表示モードは、おもに低解像度モードで漢字を表示するために設けられたモードです。アンダーラインの表示はできません。

(3) 40(80)×20 行 表示モード

高解像度・低解像度モードの両方にあるモードです。

この表示モードは、アンダーラインを表示させるために設けられたモードで、テキスト 1 行あたりのラスタ数を表 4-7 のように増やしています。この増えた部分が、アンダーライン表示用のスペースになります。

	40(80)×25 行	40(80)×20 行	40(80)×12 行	40(80)×10 行
低解像度モード	8 ラスタ／行	10 ラスタ／行	16 ラスタ／行	20 ラスタ／行
高解像度モード	16 ラスタ／行	20 ラスタ／行	32 ラスタ／行	—

表4-7 テキスト 1 行あたりのラスタ数

このモードでは縦 2 倍文字が正常に表示されません。漢字の表示は、高解像度モードのときのみ可能です。グラフィック画面は表示されませんが、グラフィック V-RAM へのアクセスは可能です。

(4) 40(80)×10 行 表示モード

低解像度モードのみにあるモードです。

この表示モードは、低解像度モードにおいて漢字と同時にアンダーラインを表示させるために設けられたモードで、テキスト 1 行あたりのラスタ数を増やしています。この増えた部分が、アンダーライン表示用のスペースになります。

このモードでは縦 2 倍文字が正常に表示されません。グラフィック画面は表示されませんが、CPU がグラフィック V-RAM にアクセスすることは可能です。

(5) 40(80)×25 行：320(640)×400 ドット 表示モード

高解像度モードのみにあるモードです。

テキスト表示は、(1) の表示モードとまったく同じです。グラフィック表示は、バンク 0 とバンク 1 を 1 ラスタごとに交互に表示させることにより解像度を 2 倍にしています。アンダーラインは表示できません。

(6) 40(80)×12 行：320(640)×384 ドット 表示モード

高解像度のみにあるモードです。

テキスト表示は、(2)の表示モードとまったく同じです。グラフィック表示は、バンク0とバンク1を交互に表示させることにより解像度を2倍にしています。アンダーラインは表示できません。

表示画面モードの設定は、CRTC 内部レジスタ、40字モード/80字モード切り換えスイッチ(8255②ポートC・ビット6)、画面管理用I/Oポートに値を設定することでおこないます。

(A) CRTC 内部レジスタ

表示画面モードを切り換える場合には、まずCRTCの内部レジスタを再設定します。本来、CRTCのR12~R15及びR16~R17は読み出すことができるようになっているのですが、X1turboではハードウェアの都合上、読み出しはできないようになっています。従って、CRTCに書き込んだ値が必要となるときは、メモリ上に値を保存しておく必要があります。以下に、CRTCの設定例を示します。

リスト4-7 CRTC設定例

CRTCAD	EQU	1800H	
CRCS1:	LD	BC, CRTCAD	
	LD	HL, CRSDT	
	XOR	A	
CRCS1:	OUT	(C), ACRTC Reg-No.指定
	INC	BC	
	LD	D, (HL)	
	OUT	(C), Dデータを書き込む
	INC	A	
	INC	HL	
	DEC	BC	
	CP	010H	
	JR	NZ, CRCS1	
	RET		
CRSDT:	DB	37H, 28H, 2DH, 34H, 0FH, 02H, 0CH, 0EH	40字モード 垂直文字数 垂直表示文字数 垂直同期位置
	DB	00H, 0FH, 60H, 07H, 00H, 00H, 00H, 00H	最大ラスタアドレス
	END		

(B) 40字モード/80字モード切り換え信号(8255②ポートC・ビット6)

X1turboでは、40字モードと80字モードで表示画面のドットクロックの周波数が違ってきます。そこで、40字モードと80字モードを切り替えるときは、40字モード/80字モード切り換え信号でこのドットクロックの周波数を切り換えます。この切り換えは、8255②ポートCに割り当てられています。表4-8がその内容ですが、このうちビット6がこのスイッチで、0にすると80字モード、1にすると40字モードになります。

・40字モード/80字モード切り換え信号

8255②ポートC・ビット6

0: 80字(640ドット)モード

1: 40字(320ドット)モード

ただし、8255②ポートCには他の信号もきているので、ポートCに直接値を書き込むと、他のビットも変化してしまい不都合な場合があります。そこで、X1turboの内部ルーチン等では、8255②の持つ、ビット・セット・リセット機能を用いて必要なビットだけを操作するようにしていま

す。これは、コントロールワードをコントロールレジスタ(I/O アドレス 1A03H)に書き込むことで、行います。

ポート	ポート端子	アクティブ	コントロール内容	信号名
C (出力)	PC7	↑	立ち上がりでプリンターは入力データをサンプルします。	STROBE
	PC6	—	80/40文字モード (H:40文字モード, L:80文字モード) 基本クロック切り換え	40/80
	PC5	↓	I/Oアクセスモード切り換え(同時アクセスモード)	GWRMD
	PC4	L	スムーズスクロール信号	スムーズスクロール
	PC3	—		—
	PC2	—		—
	PC1	—		—
	PC0	—	カセットテープへの書き込みデータ	WRITE DATA

表4-8 8255②ポートCの内容

リスト4-8 40字モードから80字モードへの切り換え例

```
CRTCAD EQU 1800H
SIOADD EQU 1A03H
S80MD: LD A, 03H
        LD HL, S8DT1
        LD BC, CRTCAD
        OUT (C), A
        INC BC
        IN A, (C)
        CP 40H
        JR C, S80M1
        LD HL, S8DT2
S80M1: LD D, 03H
        LD E, 00H
S80M2: DEC BC
        OUT (C), E .....CRTC-Reg-No指定
        LD A, (HL)
        INC BC
        OUT (C), A .....データの送信
        INC HL
        INC E
        DEC D
        JR NZ, S80M2
        LD BC, SIOADD
        LD A, 0CH
        OUT (C), A .....ポートC, ビット6をリセットする
        RET
;
S8DT1: DB 6FH, 50H, 59H, 38H .....200ライン用データ
S8DT2: DB 6BH, 50H, 59H, 88H .....400ライン用データ
;
END
```

現在200ラインか400ラインかを調べる

Dにデータ数, EにReg-No

ビデオ制御回路の基本クロックを80文字モードに

※40文字↔80文字の切り換えはCRTCの最初の4つのRegの書き換えと、ビデオ制御回路の基本クロックの変更(8255②ポートCビット6)行えばよい

リスト4-9 80字モードから40字モードへの切り換え例

```

CRTCAD EQU      1800H
SIOADD EQU      1A03H
S40MD:  LD       A, 03H
        LD       HL, S4DT1
        LD       BC, CRTCAD
        OUT      (C), A
        CP       40H
        JR       C, S40M1
        LD       HL, S4DT2
S40M1:  LD       D, 03H
        LD       E, 00H
S40M2:  DEC      BC
        OUT      (C), E .....Reg-No指定
        LD       A, (HL)
        INC      BC
        OUT      (C), A .....データ送信
        INC      HL
        INC      E
        DEC      D
        JR       NZ, S40M2
        LD       BC, SIOADD
        LD       A, 0DH
        OUT      (C), A
        RET
;
S4DT1:  DB       37H, 28H, 2DH, 34H
D4DT2:  DB       35H, 28H, 2DH, 84H
;
END

```

現在の表示モードが200ラインか400ラインか調べる

Dにデータ数

ビデオ回路の基本クロックを40文字モードに

(C) 画面管理用I/Oポート

X1シリーズには、数多くの表示画面モードを管理するための、画面管理用I/Oポート(I/Oアドレス 1FD * H)があります。表4-1がその内容ですが、このうち表示画面切り換えに関係があるのは、ビット0(低解像度/高解像度の切り換え)、ビット1(400ドット/200ドットの切り換え)、ビット2(25(20)行/12(10)行の切り換え)、ビット7(アンダーライン表示モードの切り換え)の4ビットです。以下それぞれについて説明します。

・ビット0……L/H Res. (低解像度/高解像度モード切り換え信号)

- 0 : 低解像度モード
- 1 : 高解像度モード

低解像度(200ライン)モードと高解像度(400ライン)モードの切り換え信号です。

・ビット1……1/2 RA(400ドット/200ドット切り換え信号)

- 0 : 400(384)ドットモード
- 1 : 200(192)ドットモード

高解像度(400ライン)モードの時のみ有効な信号で、400(384)ドットと200(192)ドットを切り換える時に使用します。200(192)ドットの時は、グラフィックデータを2ラスタずつ繰り返して表示させます。ただし、キャラクタ画面には影響ありません。低解像度(200ライン)モードのときは無視されます。

• ビット 2……25 / 12 行(25(20)行 / 12(10)行モード切り換え信号)

0 : 25(20)行モード

1 : 12(10)行モード

テキスト画面に対する信号です。

• ビット 7……25(12) / 20(10)行(アンダーライン表示モード切り換え信号)

0 : アンダーライン非表示モード(25(12)行モード)

1 : アンダーライン表示モード(20(10)行モード)

CRTC をアンダーライン表示モード(20(10)行モード)に設定した時に 1 にします。この時グラフィック画面は表示されません。

画面管理用の I/O ポートは書き込み専用ポートで、読み出すことはできませんが、BIOS ROM 内のルーチンはメイン RAM の F8D6H 番地をバッファとして、書き込んだ値を保存しています。したがって、BIOS ROM 内ルーチンを使用して画面モード等を切り換えた場合、F8D6H 番地を参照することで画面管理用 I/O ポートの設定値を知ることができます。

X1turbo の各表示画面モードでのそれぞれの値と、表示画面設定プログラムの例を以下に示します。

• 低解像度モード

テキスト画面		40×25	80×25	40×12	80×12	40×20	80×20	40×10	80×10
グラフィック画面		320×200	640×200	320×192	640×192				
レジスタ									
C R T C 内部 レジスタ	R 0	3 7	6 F	3 7	6 F	3 7	6 F	3 7	6 F
	R 1	2 8	5 0	2 8	5 0	2 8	5 0	2 8	5 0
	R 2	2 D	5 9	2 D	5 9	2 D	5 9	2 D	5 9
	R 3	3 4	3 8	3 4	3 8	3 4	3 8	3 4	3 8
	R 4	1 F	1 F	0 F	0 F	1 8	1 8	0 B	0 B
	R 5	0 2	0 2	0 2	0 2	0 8	0 8	1 2	1 2
	R 6	1 9	1 9	0 C	0 C	1 4	1 4	0 A	0 A
	R 7	1 C	1 C	0 E	0 E	1 6	1 6	0 B	0 B
	R 8	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
	R 9	0 7	0 7	0 F	0 F	0 9	0 9	1 3	1 3
	R 10	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
	R 11	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
	R 12	↓	↓	↓	↓	↓	↓	↓	↓
	R 13	↓	↓	↓	↓	↓	↓	↓	↓
	R 14	↓	↓	↓	↓	↓	↓	↓	↓
	R 15	↓	↓	↓	↓	↓	↓	↓	↓
8255② ポートC	B 6	1	0	1	0	1	0	1	0
画 面 管 理 用 I/O ポ ー ト	DB 0	0	0	0	0	0	0	0	0
	DB 1	0	0	0	0	—	—	—	—
	DB 2	0	0	1	1	0	0	1	1
	DB 3	—	—	—	—	—	—	—	—
	DB 4	—	—	—	—	—	—	—	—
	DB 5	—	—	—	—	—	—	—	—
	DB 6	—	—	—	—	—	—	—	—
	DB 7	0	0	0	0	1	1	1	1

• 高解像度モード

テキスト画面		40×25	80×25	40×12	80×12	40×25	80×25	40×12	80×12	40×20	80×20
グラフィック画面		320×200	640×200	320×192	640×192	320×400	640×400	320×384	640×384		
レジスタ											
CRTC内部レジスタ	R0	3 5	6 B	3 5	6 B	3 5	6 B	3 5	6 B	3 5	6 B
	R1	2 8	5 0	2 8	5 0	2 8	5 0	2 8	5 0	2 8	5 0
	R2	2 D	5 9	2 D	5 9	2 D	5 9	2 D	5 9	2 D	5 9
	R3	8 4	8 8	8 4	8 8	8 4	8 8	8 4	8 8	8 4	8 8
	R4	1 B	1 B	0 D	0 D	1 B	1 B	0 D	0 D	1 5	1 5
	R5	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 8	0 8
	R6	1 9	1 9	0 C	0 C	1 9	1 9	0 C	0 C	1 4	1 4
	R7	1 A	1 A	0 D	0 D	1 A	1 A	0 D	0 D	1 5	1 5
	R8	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
	R9	0 F	0 F	1 F	1 F	0 F	0 F	1 F	1 F	1 3	1 3
	R10	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
	R11	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
	R12										
	R13										
	R14										
	R15										
8255②ポートC	B 6	1	0	1	0	1	0	1	0	1	0
画面管理ポート	DB 0	1	1	1	1	1	1	1	1	1	1
	DB 1	1	1	1	1	0	0	0	0	—	—
	DB 2	0	0	1	1	0	0	1	1	0	0
	DB 3	—	—	—	—	—	—	—	—	—	—
	DB 4	—	—	—	—	—	—	—	—	—	—
	DB 5	—	—	—	—	—	—	—	—	—	—
	DB 6	—	—	—	—	—	—	—	—	—	—
	DB 7	0	0	0	0	0	0	0	0	1	1

表4-9 各表示モードでのCRTC内部レジスタ、40字／80字モード切り換え信号、画面管理I/Oポートの設定値

リスト4-10 表示画面設定例 1

```

CRTCAD EQU      1800H
SIOADD EQU      1A03H
SCRNIO EQU      1FDOH
SMDA1: LD        BC, CRTCAD
      LD         HL, SMDAD
      XOR        A
SMSAD: OUT        (C), A
      INC        BC
      LD         D, (HL)
      OUT        (C), D
      INC        A
      INC        HL
      DEC        BC
      CP         10H
      JR         NZ, SMDA1
      LD         A, (HL)
      LD         BC, SCRNIO
      OUT        (C), A .....画面管理ポートへ出力
      INC        HL
      LD         A, (HL)
      LD         BC, SIOADD
      OUT        (C), A .....基本クロックの設定
      RET
;
SMDAS: DB        6FH, 50H, 59H, 38H, 1FH, 02H, 19H, 1CH
      DB         00H, 07H, 60H, 07H, 00H, 00H, 00H, 00H
      DB         20H .....画面管理I/Oポート参照
      DB         0CH .....80文字モード
      ;
      END

```

CRTCを80×25文字モードでイニシャライズ


```

CRTCAD EQU      1800H
SIOADD EQU      1A03H
SCRNIO EQU      1FD0H
SMDBS: LD        BC, CRTCAD
      LD        HL, SWDBD
      XOR       A
SMDB1: OUT       (C), A
      INC      BC
      LD        D, (HL)
      OUT      (C), D
      INC      A
      INC      HL
      DEC      BC
      CP       010H
      JR       NZ, SMDB1
      LD        A, (HL)
      LD        BC, SCRNIO
      OUT      (C), A .....画面管理ポートへ出力
      INC      HL
      LD        A, (HL)
      LD        BC, SIOADD
      OUT      (C), A .....基本クロックの設定
      RET
;
SMDBD: DB        37H, 28H, 2DH, 34H, 0FH, 02H, 0CH, 0EH
      DB        00H, 0FH, 60H, 07H, 00H, 00H, 00H, 00H
      DB        64H .....画面管理I/Oポート参照
      DB        0DH .....40文字モード
      ;
      END

```

CRTCを40×12文字モードでイニシャライズ

4-2-4 表示画面ページの切り換え

Xlturboは、2000文字分のテキストV-RAMと640×400ドットで8色表示のできる、グラフィックV-RAMを1画面分装備しています。表示画面モードによっては、これらのV-RAMをいくつか分割して使用することができます。このとき、それぞれの領域をページと呼びます。

• 低解像度モード

テキスト・モード	40×25	80×25	40×12	80×12	40×20	80×20	40×10	80×10
グラフィック・モード	320×200	640×200	320×192	640×192				
テキスト・ページ	2	1	2	1	2	1	2	1
グラフィック・ページ	4	2	4	2				

• 高解像度モード

テキスト・モード	40×25	80×25	40×12	80×12	40×25	80×25	40×12	80×12	40×20	80×20
グラフィック・モード	320×200	640×200	320×192	640×192	320×400	640×400	320×384	640×384		
テキスト・ページ	2	1	2	1	2	1	2	1	2	1
グラフィック・ページ	4	2	4	2	2	1	2	1		

表4-10 各表示画面に対するテキスト画面とグラフィック画面のページ数

ページ切り換えは、CRTC 内部レジスタの R12 と画面管理 I/O ポートのビット 3 で行います。

(1) CRTC 内部レジスタ・R12

R12(上位スタートアドレスレジスタ)は、CRTC が出力するリフレッシュメモリアドレス (MA) の先頭アドレスの上位バイトを設定します。表 4-11 に各表示画面モードにおけるページ の先頭アドレスを示します。ただし、実際にこのレジスタに設定される値は、V-RAM 先頭番地 からのオフセットになります。

また、Xlturbo では、1 個の CRTC でテキスト画面とグラフィック画面の両方を制御している ので、この場合にはテキストページとグラフィックページが同時に切り換わります。

(1) テキスト画面

ページ \ モード	40×25	80×25	40×12	80×12	40×20	80×20	40×10	80×10
ページ 0	3000H	3000H	3000H	3000H	3000H	3000H	3000H	3000H
ページ 1	3400H		3200H		3400H		3200H	

(2) グラフィック画面

ページ \ モード	320×200	640×200	320×192	640×192	320×400	640×400	320×384	640×384
ページ 0	4000H	4000H	4000H	4000H	4000H	4000H	4000H	4000H
ページ 1	4400H	4000'H	4200H	4000'H	4400H		4200H	
ページ 2	4000'H		4000'H					
ページ 3	4400'H		4200'H					

表4-11 各表示画面モードでのテキストページ・グラフィックページ の先頭アドレス

この表は、テキスト V-RAM の表示先頭アドレスを示しています。漢字用テキスト V-RAM の アドレスは、(テキスト V-RAM のアドレス)+800H となります。アトリビュート V-RAM のア ドレスは、(テキスト V-RAM のアドレス)-1000H です。

グラフィック画面については、BLUE 用 V-RAM の先頭アドレスを示してあります。RED 用 V-RAM、GREEN 用 V-RAM のアドレスは、それぞれ(BLUE 用 V-RAM アドレス)+4000H、 (BLUE 用 V-RAM アドレス)+8000H になります。なお、ダッシュマークのついたアドレスは BANK1 におけるアドレスを示します。

(2) 画面管理用 I/O ポート・ビット 3

グラフィック画面に対する信号で、画面上にグラフィック V-RAM バンク 0、バンク 1 のどち らかを表示するかを選択します。

0：グラフィック V-RAM バンク 0 を表示

1：グラフィック V-RAM バンク 1 を表示

この信号は、テキスト画面には影響しません。テキストページとは独立にグラフィックペー ジのみ切り換わります。

次に各画面モードにおけるそれぞれの設定値と、ページ切り換えのプログラム例を示します。

リスト4-12 ページ切り換えプログラム例

```
CRTCAD EQU 1800H
PCDATA EQU 4 .....ページ1を選択
SCRNIO EQU 1FD0H
WK1FDO EQU 0F8D6H
PCNGE: LD BC, CRTCAD
      LD A, 0CH .....CRTC Reg-No#
      OUT (C), A
      INC BC
      LD A, PCDATA
      OUT (C), A
      LD BC, SCRNIO
      LD A, (WK1FDO)
      AND 07H
      OUT (C), A ] ディスプレイページを1に
      RET
      ;
      END
```

モード		40×25	80×25	40×12	80×12	40×25	80×25	40×12	80×12	40×20	80×20	40×10	80×10
		320×200	640×200	320×192	640×192	320×400	640×400	320×384	640×384				
ページ 0	CRTC R12	00H	00H	00H	00H	00H	00H	00H	00H	00H	00H	00H	00H
	1FD0H D3	0	0	0	0	—	—	—	—	—	—	—	—
ページ 1	R12	04H	00H	02H	00H	04H		02H		04H		02H	
	D 3	0	1	0	1	—		—		—		—	
ページ 2	R12	00H		00H									
	D 3	1		1									
ページ 3	R12	04H		02H									
	D 3	1		1									

表4-12 各画面モードでのページ切り換え設定値

4-3 テキスト画面

X1 シリーズは、テキスト画面にキャラクタ・ジェネレータ(CG)，プログラマブル・キャラクタ・ジェネレータ(PCG)，漢字 ROM の3種類のフォントを表示することができます。(X1 ではCG と PCG のみ。漢字はグラフィック V-RAM 上に書き込まれる。)

(1) キャラクタ・ジェネレータ(CGROM)：8K バイト(X1 は 4K バイト)

通常使用する英数字，カタカナ，グラフィックパターン等を納めた ROM で，前半の 4K バイトに 8×8 ドット構成の文字を256文字分，後半の 4K バイトに 8×16ドット構成の文字を256文字分納めてあります。(X1 は 8×8 ドットのパターンのみ)

(2) プログラマブル・キャラクタ・ジェネレータ(PCGRAM)：2K バイト×3

CG，漢字 ROM が読み出し専用なのに対して，PCG はユーザーが自由にフォントを定義して使用できるキャラクタ・ジェネレータです。PCG は，BLUE，RED，GREEN の各色について 2K バイトずつ，合計 6K バイトあり，ドットごとに 8 色で，8×8 ドット構成で256個，8×16ドット構成で128個のキャラクタを定義することができます。(X1 は 8×8 ドットのみ)

(3) 漢字 ROM：第 1 水準 128K バイト，第 2 水準 128K バイト

JIS 非漢字 453 字と JIS 第 1 水準漢字 2965 字，及び JIS 第 2 水準 3384 字を，1 文字 16×16 ドット構成で納めた ROM です。フォントデータは，2 組の ROM に，漢字の右半分と左半分に分けて書き込まれています。

4-3-1 フォント表示

テキスト画面に，どのフォントデータを表示するかは，その画面表示位置に対する V-RAM，漢字テキスト V-RAM，アトリビュート V-RAM の内容で決まります。この項では，CG，PCG，漢字 ROM のフォントデータが，表示の時にどのように選択されるかを説明します。

まず，CG，PCG，漢字 ROM のうち，どれを表示させるかを指定します。これは，漢字用テキスト V-RAM とアトリビュート V-RAM のなかの次の 3 つのビットの情報によって選択されます。

- ①アトリビュート V-RAM・ビット 5：ROM/RAM 選択信号
- ②漢字用テキスト V-RAM・ビット 4：漢字第 1/2 水準選択信号
- ③漢字用テキスト V-RAM・ビット 7：CGROM/漢字 ROM 選択信号

それぞれのビットの値と表示されるフォントは，以下のようになります。

ROM/RAM	CG/KANJI	1/2 水準	テキスト表示
0	0	*	CG
0	1	0	漢字 ROM (第 1 水準)
0	1	1	漢字 ROM (第 2 水準)
1	0	0	PCG キャラクタ方式
1	0	1	PCG 外字方式(①)
1	1	*	PCG 外字方式(②)

表4-13 ROM/RAM信号，CG/漢字信号，1/2水準信号の値と表示フォント

- ・PCG キャラクタ方式は，PCG のデータを 8×8 ドット構成で扱います。
- ・PCG 外字方式では，PCG のデータを 8×16 ドット構成で扱います。
- ・PCG 外字方式①と PCG 外字方式②とは，ハードウェア的にはまったく同じです。この区別は，ソフトウェア上で識別するために設けられたもので，例えば BASIC では，PCG 外字方式①を半角文字(8×16 ドット構成)，PCG 外字方式②を全角文字(16×16 ドット構成)を表示させるのに割り当てています。

CG，PCG，漢字 ROM のうち 1 つを選択した上で，選択したフォント ROM/RAM 上でのアドレスを決定する必要があります。以下，それぞれについて説明します。

(1) キャラクタ・ジェネレータ(CGROM)

ROM/RAM 信号と CG/KANJI 信号がともに“0”のときに，この CGROM が画面表示用として選択されます。このとき，テキスト V-RAM には ASCII コードをいれておきます。

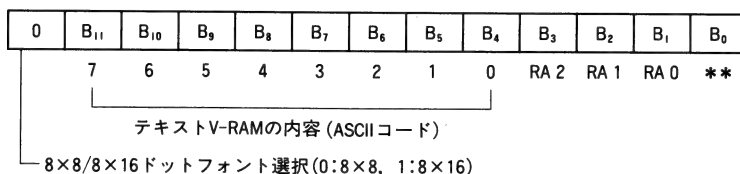
フォントは，低解像度の 25 行モードと 20 行モードのときのみ，8×8 ドット構成のフォントが使用され，その他の表示画面モードでは，8×16 ドット構成のフォントが使用されます。この切り換えは，画面管理用 I/O ポートの値によって，自動的に行なわれます。I/O ポートの値と表示されるフォントの関係をまとめると，次のようになります。

L/H Res.	25/12行	フォント
0	0	8×8ドット
0	1	8×16ドット
1	0	//
1	1	//

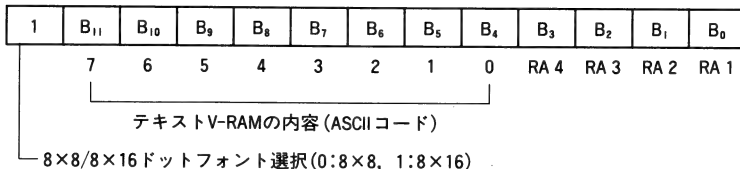
表4-14 L/H Res.信号, 25(20)/12(10)行信号の値と表示されるCGフォント

CGROM に与えられるアドレスの内容は、次のようになっています。ここで RA0~RA3 は、CRTC が発生するラスタアドレスです。

●低解像度・25(20)行モード



●高解像度・12行モード



●その他の表示画面モード

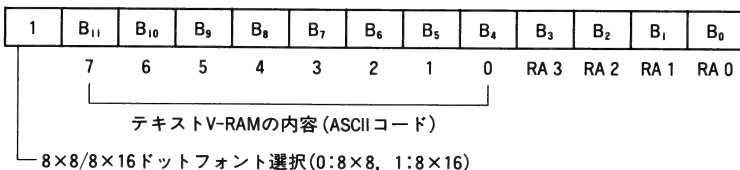


図4-16 CGROMに与えられるアドレスの内容

(2) 漢字 ROM

漢字テキスト V-RAM の CG/KANJI 信号が “1” で、アトリビュート V-RAM の ROM/RAM 信号が “0” のときには漢字 ROM が選択されます。このとき、テキスト V-RAM には漢字 ROM アドレスの下位 8 ビットを、漢字用テキスト V-RAM のビット 0～3 に漢字 ROM アドレスの上位 4 ビットを格納しておきます。

漢字 ROM は、第一水準と第二水準それぞれ 4 個の、計 8 個から構成されています。それぞれの ROM は、No.0 と No.2 が漢字の左半分、No.1 と No.3 が右半分のフォントを格納しており、0 と 1, 2 と 3 が組になって 16×16ドットの漢字を表示しています。

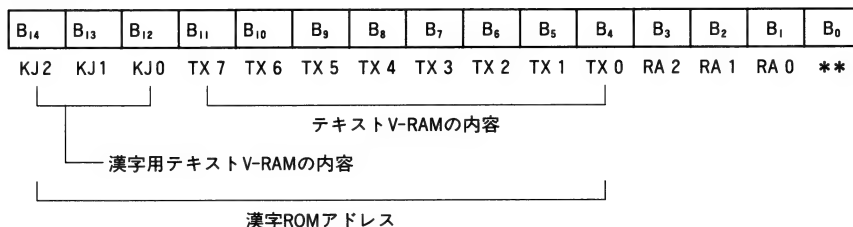
この 8 個の ROM から 1 つを選択するわけですが、それには第 1 水準/第 2 水準選択信号(漢字用テキスト V-RAM・ビット 4), LEFT/RIGHT 選択信号(漢字用 V-RAM・ビット 6), 漢字 ROM アドレスの最上位ビット(漢字用テキスト V-RAM・ビット 3)の 3 つの信号を使います。それぞれの信号の値と選択される漢字 ROM の関係は次の通りです。

1 / 2 水準	LEFT/RIGHT	ASCII 2・ビット3	漢字ROM
0	0	0	第1水準No.0
0	1	0	第1水準No.1
0	0	1	第1水準No.2
0	1	1	第1水準No.3
1	0	0	第2水準No.0
1	1	0	第2水準No.1
1	0	1	第2水準No.2
1	1	1	第2水準No.3

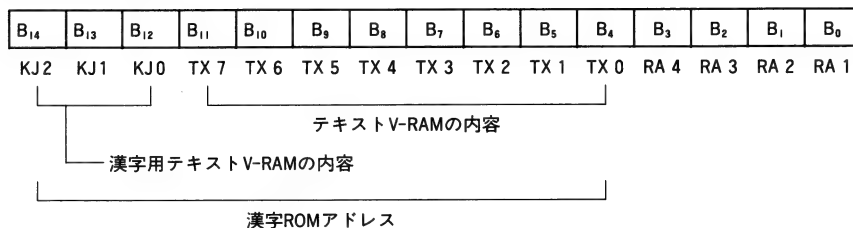
表4-15 漢字ROMの選択

漢字ROMに与えられる15ビットのアドレスの内容を、次に示します。

●低解像度・25(20)行モード



●高解像度・12行モード



●その他の表示画面モード

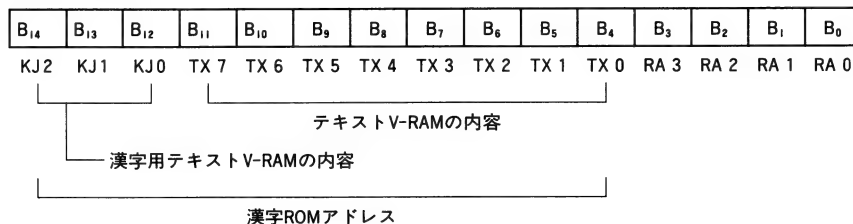


図4-17 漢字ROMに与えられるアドレスの内容

※低解像度の25行モード、20行モードでは、漢字は正常に表示されません。

(3) プログラマブル・キャラクタ・ジェネレータ(PCGRAM)

アトリビュート V-RAM の ROM/RAM 信号が "1" のときには、プログラマブル・キャラクタ・ジェネレータ(PCGRAM)が選択されます。このとき、テキスト V-RAM には ASCII コードをいれておきます。

PCGRAM は、BLUE, RED, GREEN の 3 つで構成されており、それらを合成して表示します。

PCG キャラクタ方式は、1文字を 8×8 ドット構成とみなして表示します。PCG 外字方式は、1文字を 8×16 ドット構成で表示します。この 2 つの方式の選択は、CG/KANJI 信号と 1/2 水準信号によってきまります。CG/KANJI 信号と 1/2 水準信号がともに "0" のときには PCG キャラクタが選択され、それ以外のときには PCG 外字方式が選択されます。

CG/KANJI	1/2水準	テキスト表示
0	0	PCG キャラクタ方式
0	1	PCG 外字方式 (①)
1	*	PCG 外字方式 (②)

表4-16 CG/KANJI信号, 1/2水準信号とPCG表示方式

PCGRAM に与えられる11ビットのアドレスが、それぞれの方式でどのように変わるかを次の図に示します。

表示モード	PCGアクセス方式	PCGに与えられるアドレス																																	
低解像度・25行モード 20行モード	PCGキャラクタ方式	<table><tr><td>B10</td><td>B 9</td><td>B 8</td><td>B 7</td><td>B 6</td><td>B 5</td><td>B 4</td><td>B 3</td><td>B 2</td><td>B 1</td><td>B 0</td></tr><tr><td>TX7</td><td>TX6</td><td>TX5</td><td>TX4</td><td>TX3</td><td>TX2</td><td>TX1</td><td>TX0</td><td>RA2</td><td>RA1</td><td>RA0</td></tr><tr><td colspan="11">└────────── テキストV-RAMの内容 ─────────┘</td></tr></table>	B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	RA2	RA1	RA0	└────────── テキストV-RAMの内容 ─────────┘										
	B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0																								
TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	RA2	RA1	RA0																									
└────────── テキストV-RAMの内容 ─────────┘																																			
低解像度・12行モード 10行モード 高解像度・25行モード 20行モード	PCGキャラクタ方式	<table><tr><td>B10</td><td>B 9</td><td>B 8</td><td>B 7</td><td>B 6</td><td>B 5</td><td>B 4</td><td>B 3</td><td>B 2</td><td>B 1</td><td>B 0</td></tr><tr><td>TX7</td><td>TX6</td><td>TX5</td><td>TX4</td><td>TX3</td><td>TX2</td><td>TX1</td><td>TX0</td><td>RA3</td><td>RA2</td><td>RA1</td></tr><tr><td colspan="11">└────────── テキストV-RAMの内容 ─────────┘</td></tr></table>	B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	RA3	RA2	RA1	└────────── テキストV-RAMの内容 ─────────┘										
	B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0																								
	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	RA3	RA2	RA1																								
	└────────── テキストV-RAMの内容 ─────────┘																																		
PCG外字方式	<table><tr><td>B10</td><td>B 9</td><td>B 8</td><td>B 7</td><td>B 6</td><td>B 5</td><td>B 4</td><td>B 3</td><td>B 2</td><td>B 1</td><td>B 0</td></tr><tr><td>TX7</td><td>TX6</td><td>TX5</td><td>TX4</td><td>TX3</td><td>TX2</td><td>TX1</td><td>RA3</td><td>RA2</td><td>RA1</td><td>RA0</td></tr><tr><td colspan="11">└────────── テキストV-RAMの内容 ─────────┘</td></tr></table>	B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	TX7	TX6	TX5	TX4	TX3	TX2	TX1	RA3	RA2	RA1	RA0	└────────── テキストV-RAMの内容 ─────────┘											
B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0																									
TX7	TX6	TX5	TX4	TX3	TX2	TX1	RA3	RA2	RA1	RA0																									
└────────── テキストV-RAMの内容 ─────────┘																																			
高解像度・12行モード	PCGキャラクタ方式	<table><tr><td>B10</td><td>B 9</td><td>B 8</td><td>B 7</td><td>B 6</td><td>B 5</td><td>B 4</td><td>B 3</td><td>B 2</td><td>B 1</td><td>B 0</td></tr><tr><td>TX7</td><td>TX6</td><td>TX5</td><td>TX4</td><td>TX3</td><td>TX2</td><td>TX1</td><td>TX0</td><td>RA4</td><td>RA3</td><td>RA2</td></tr><tr><td colspan="11">└────────── テキストV-RAMの内容 ─────────┘</td></tr></table>	B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	RA4	RA3	RA2	└────────── テキストV-RAMの内容 ─────────┘										
	B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0																								
	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	RA4	RA3	RA2																								
	└────────── テキストV-RAMの内容 ─────────┘																																		
PCG外字方式	<table><tr><td>B10</td><td>B 9</td><td>B 8</td><td>B 7</td><td>B 6</td><td>B 5</td><td>B 4</td><td>B 3</td><td>B 2</td><td>B 1</td><td>B 0</td></tr><tr><td>TX7</td><td>TX6</td><td>TX5</td><td>TX4</td><td>TX3</td><td>TX2</td><td>TX1</td><td>RA4</td><td>RA3</td><td>RA2</td><td>RA1</td></tr><tr><td colspan="11">└────────── テキストV-RAMの内容 ─────────┘</td></tr></table>	B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	TX7	TX6	TX5	TX4	TX3	TX2	TX1	RA4	RA3	RA2	RA1	└────────── テキストV-RAMの内容 ─────────┘											
B10	B 9	B 8	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0																									
TX7	TX6	TX5	TX4	TX3	TX2	TX1	RA4	RA3	RA2	RA1																									
└────────── テキストV-RAMの内容 ─────────┘																																			

表4-17 PCGRAMに与えられるアドレスの内容

PCG キャラクタ方式では、 8×8 ドットのフォントを256種類持つことができます。

PCG 外字方式では、テキスト V-RAM からのデータが7ビット、ラスタアドレスが4ビットとなり、 8×16 ドットのフォントを128種類持つことができます。この時、テキスト V-RAM の ASCII コードの最下位ビットは無視されます。連続した2つの ASCII コードを持つ PCG パターンのうち、偶数の ASCII コードを持つ 8×8 ビットのパターンが上半分、奇数の ASCII コードのパターンが下半分となって、 8×16 ドットのフォントを構成します。テキスト V-RAM に書き込まれる ASCII コードには、偶数・奇数の区別がなくなり、偶数・奇数の順で連続した2つの ASCII コードは、どちらも同じ1つの 8×16 ドットのフォントを表示させます。

例えば ASCII コード 42H の PCG に[B]のフォント、43H に[C]のフォントが入っていたとすると、PCG キャラクタ方式と、PCG 外字方式の表示は、それぞれ図4-21のようになります。

PCG 外字方式は、漢字表示や $1/4$ 角表示に応用することができます。 8×16 ドットのフォントを2つ使って1つに漢字の左側部分、もう1つに右側部分を定義しておいて、並べて表示すれば 16×16 ドットの漢字を表示することができますし、 8×16 ドットのパターンのうち半分を空白にしておけば $1/4$ 角文字を表示させることもできます。この表示例を下図に示します。

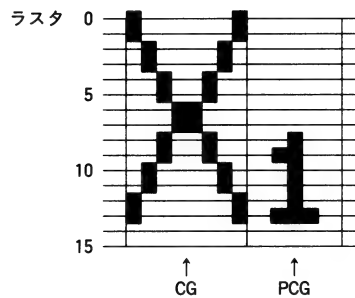


図4-18 PCG外字方式による $1/4$ 角文字の表示例

PCG 外字方式は、1行あたり16ラスタ以上のラインが必要ですので、低解像度の25行モード、20行モードでは使用することができません。ただし、縦倍表示にすれば表示可能です。

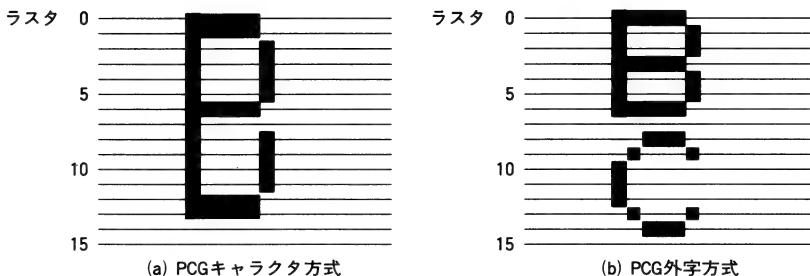


図4-19 PCGキャラクタ方式とPCG外字方式による画面表示例

以下に、各画面表示モードにおけるテキストの表示例を示します。

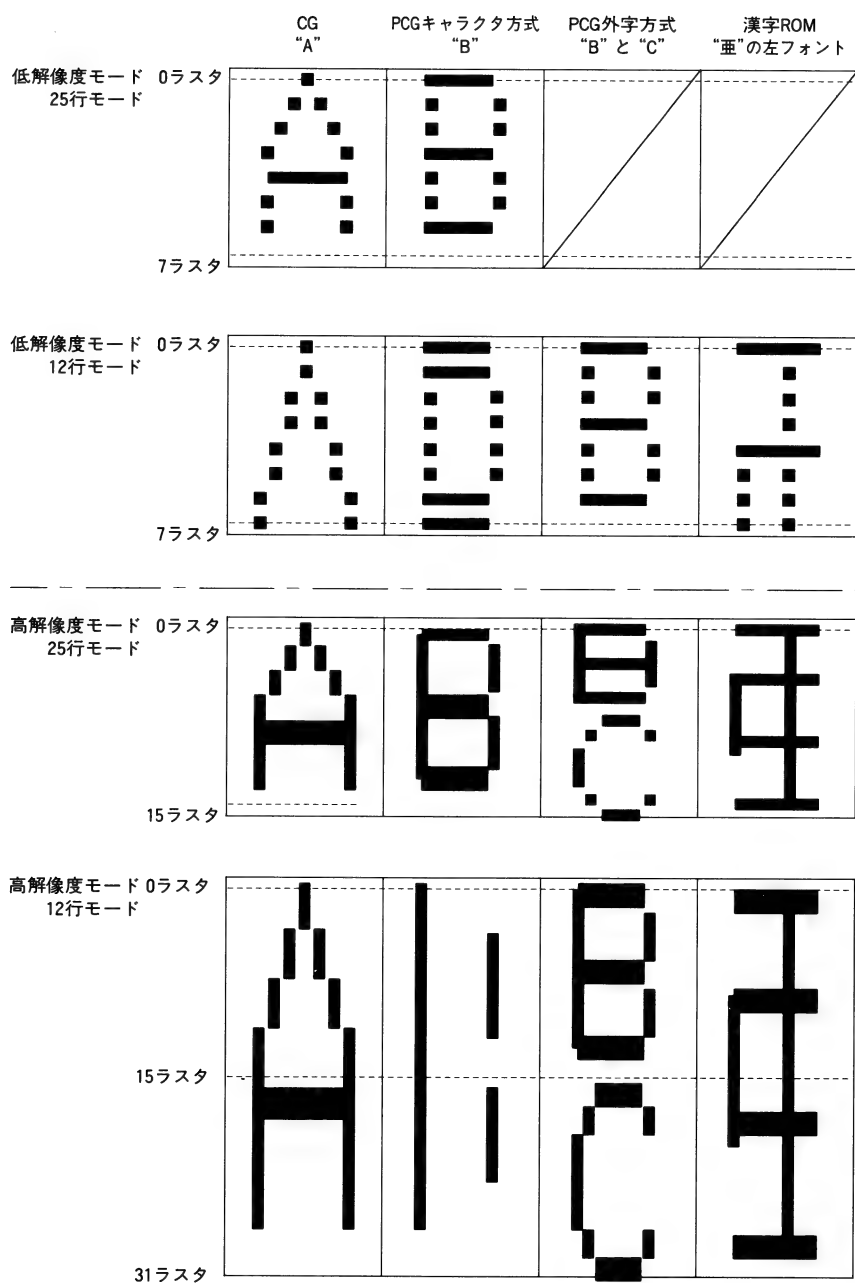


図4-20

4-3-2 CPUからのフォントのアクセス

CG・PCG・漢字ROMが表示に使用されているときに、それらをCPUからアクセスすると、画面表示のアクセスと競合し、画面にノイズがでて見にくくなってしまいます。そこで、一般的には、画面が表示されないとき(帰線期間)に、これらに対してアクセスする方法が使われます。

X1シリーズでも、この方法が使われていますが、帰線期間の利用法の違い等により、通常アク

セスモードと高速アクセスモード(turboのみ)の2種類のモードに分かれています。

通常アクセスモードは、X1全シリーズで使われている方式で、垂直帰線期間のみを使用します。このアクセスモードでは、PCGの256キャラクタすべてを設定するのに、約12秒もかかってしまいます。また、垂直帰線期間の始まりの検知や、タイミングの制御をソフトウェアで行っているため、ソフトウェアに負担がかかります。その上X1turboのように表示画面モードが多い場合には、表示モードごとにアクセスタイミングの微調整をしなければならないので、ソフトウェア的な負担は、さらに大きなものとなってしまいます。

そこで、X1turboには、高速アクセスモードが設けられています。高速アクセスモードは、垂直帰線期間ではなく水平帰線期間を、CG等のアクセスに使用するアクセスモードです。通常アクセスモードでは、垂直帰線1回で8バイトをアクセスできるので、画面表示1フレームあたり、8バイトのアクセスが可能です。これに対して、高速アクセスモードは、水平帰線期間1回に1バイトをアクセスします。水平帰線期間は1フレームあたり、低解像度モードで200回、高解像度モードで400回あるので、1フレームあたりのアクセスバイト数は、200バイト又は400バイトとなり、通常アクセスモードと比較して単純計算でも8～45倍と、アクセス速度が格段に速くなっています。さらに、帰線期間の検知やアクセスタイミングの制御をハードウェアで行っているため、タイミングに関するソフトウェアの負担がほとんどなくなっています。

どちらのアクセスモードを使用するかは、画面管理用I/Oポート(I/Oアドレス 1FD * H)のビットSPCG/FPCGによって指定されます。

4-3-3 通常アクセスモード

X1全シリーズで利用可能で、X1とturboシリーズでコンパチビリティを保つために設けられているアクセスモードです。CGROM・PCGRAMに対してのみアクセス可能で、漢字ROMに対してはアクセスすることはできません。以下に、通常アクセスモードの原理を説明します。

テキストV-RAMは、それぞれ2048バイトずつ設けられていますが、そのすべての表示用に使われるわけではありません。40字モードの場合には、最初の1024バイトを0ページ、のこりの1024バイトを1ページと2つにわけて使っていますが、表示用に使われているのはそれぞれ先頭から、 $40 \times 25 = 1000$ バイトでのこりの24バイトは表示されません。80字モードの場合にも、同じように2048バイトのうち表示されるのは、先頭から $80 \times 25 = 2000$ バイトで、のこりの48バイトは表示されません。(図4-6参照)

しかし、この表示されない部分に入ってもCRTCは、あたかも表示部分がつづいているかのようにメモリアドレス(MA)・ラスタアドレス(RA)を発生しつづけます。一方、表示用のフォント選択回路等も正常に動作しつづけているので、メモリアドレスによって指定された各テキストV-RAMの内容とラスタアドレスから、まえの「フォント表示」のところで説明した規則に従ってCGROM等からアクセスされ、そのデータライン上にフォントデータが出力されます。CGROM等のデータラインは、CPUのI/Oアドレス上にもマッピングされているので、このタイミングをみはからって、I/Oアドレスを読み込めばフォントデータを取り込むことができます。PCGに対しては、書き込みもできます。

CGROM等のデータラインがマッピングされているI/Oアドレスは、以下のとおりです。

CGROMI/O アドレス	14**H
PCGRAM・BLUEI/O アドレス	15**H
PCGRAM・REDI/O アドレス	16**H
PCGRAM・GREENI/O アドレス	17**H

次に、読み込み／書き込みのタイミングについて説明します。
まず、最初に垂直帰線期間の始まりを知らなければなりません。これは、8255②ポートB・ビット7 (V-DISP 信号)の立ち下がり(1から0への変化点)によって知ることができます。

ポート	ポート端子	アクティブ	コントロール内容	信号名
B (入力)	PB ₇	L	垂直帰線期間信号	V-DISP
	PB ₆	H	データ転送禁止信号	IBF
	PB ₅	L	80C49からのデータ受信可能指示信号	OBF
	PB ₄	H	BIOS ROMバンク切り換え信号 (L: BIOS ROM側, H: メインメモリ側)	IPL RESET
	PB ₃	L	プリンターからの入力可能指示信号	BUSY
	PB ₂	H	垂直同期信号	PV. SYNC
	PB ₁	—	カセット読み出しデータ	READ. DATA
	PB ₀	L	BREAK信号	BREAK

表4-18 8255②ポートB(I/Oアドレス1A01H)の内容

- ・ビット7……V-DISP(垂直帰線期間信号)
 - 0：非表示(垂直帰線)期間
 - 1：表示期間

したがって、このビットが1から0に変化した時が垂直帰線期間の始まりとなります。
ここから、8 ラスタの間フォントデータの順次読み込み／書き込みが可能になります。しかし、実際に有効なのは、表示されない部分の V-RAM がフォントを指定できる、各ラスタの最初の 24μsec に限られます。各ラスタの始まりは、直接知る方法がないので、垂直帰線期間の始まりからソフトウェア的にタイミングを合わせていきます。
各ラスタでは、ラスタの始まりから順次、V-RAM のデータを使って CGROM 等のアドレスを指定していきますので、同じ ASCII コードを書き込んでおけば最大で 24μsec の間、1つのアドレスを指定しておくことができます。この間に、CPU から I/O アドレスをアクセスしなければならないのですが、1バイトのデータをアクセスするには十分な時間です。プログラムによっては、PCGRAM の BLUE, RED, GREEN, それぞれ1ライン分、計3バイトのアクセスも可能です。以下に、アクセスタイミング図を示します。

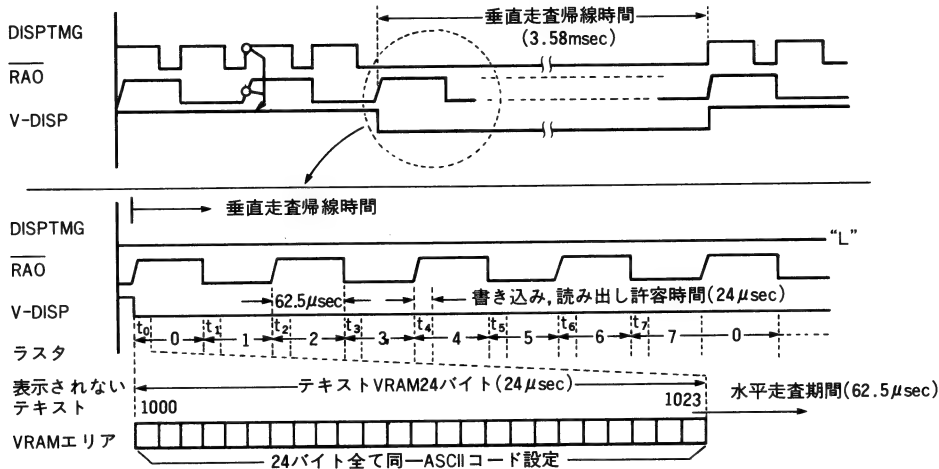


図4-21 PCGへのアクセスタイミング図

以上のことから、通常アクセスモードのソフトウェア手順をまとめると以下のようになります。

- (1) テキスト V-RAM の表示されない部分に、アクセスしようとするフォントの ASCII コードを24バイト書き込んでおきます。24バイトはすべて同じコードでなければなりません。
- (2) 漢字用テキスト V-RAM, アトリビュート V-RAM の該当部分に、PCGRAM と CGROM のうちどちらをアクセスするか情報を(1)と同じく24バイト書き込みます。PCGRAM と CGROM の選択は「フォントアクセス」の項で説明した規則に従います。したがって、漢字用テキスト V-RAM のビット 7 (CG/KANJI 信号)と、アトリビュート V-RAM のビット 5 (ROM/RAM 信号)によって決定されます。他のビットの情報は、無関係です。なお、通常アクセスモードでは、漢字 ROM をアクセスすることはできません。
- (3) 画面管理用 I/O ポート (I/O アドレス 1FD * H) のビット 5 (SPCG/FPCG) 信号を 0 に設定します。これで通常アクセスモードになります。
- (4) 8255②ポート B (I/O アドレス 1A01H) のビット 7 (V-DISP 信号)を読み込んで、1 から 0 に変わるまで待ちます。この変化点 (立ち下がり) が、垂直帰線期間の始まりとともに第 0 ラスタの開始を表します。
- (5) CGROM, PCGRAM のフォントデータを I/O アドレスを通じて読み出し / 書き込みます。
- (6) つぎのラスタの開始まで、ソフトウェア的に待ちます。1 ラスタあたりの時間は、約 62.6μsec ですから、ここから残り時間を計算して待ちます。
- (7) (5) と (6) を 8 ラスタ分繰り返します。

※垂直表示期間 (V-DISP 信号 “1”) に CRTIC の内部レジスタの内容を変更すると、その直後の帰線期間 (V-DISP 信号 “0”) のメモリアドレスが正常に出力されません。従って、CRTIC の内部レジスタの内容を変更した場合、16msec 以上待ってから CGROM 等へのアクセスを行なわなければなりません。垂直帰線期間中に CRTIC の内部レジスタの内容を変更したあと、CGROM 等へのアクセスを行うようにプログラムすれば、CGROM 等へのアクセスルーチンで次の V-DISP 信号の立ち下がりまで待つので、自動的に 16msec 以上の時間余裕が取られます。

4-3-4 高速アクセスモード

高速アクセスモードは、X1turbo のみのアクセスモードで、すべての表示画面モードで使うことができます。

高速アクセスモードは、通常アクセスモードが、CGROM 等へアクセスするのに垂直帰線期間を使用するのに対して、水平帰線期間をアクセスに使用し、CGROM 等へのアクセスを通常アクセスモードと比較して約 8 ~ 45 倍の速度で行うことができます。また、帰線期間の検知やアクセスタイミング等の制御をハードウェアで行っているのでソフトウェアの負担がほとんどなくなっています。

以下に、高速アクセスモードの原理を説明します。

まず、CGROM 等がマッピングされている I/O ポート (I/O アドレス 14 * * H ~ 17 * * H) をアクセスすると、ハードウェアによって CPU にウェイトがかけられます。この時点が、高速アクセスモードでのアクセス開始点になります。水平帰線期間に入ると、いままで CRTIC からメモリアドレスが使用されていた、各テキスト V-RAM へのアドレスが、各テキスト V-RAM の最上位の 1 バイトを示すように切り換えられます。つまり、テキスト V-RAM では 37FFH、漢

字用テキスト V-RAM では 3FFFH, アトリビュート V-RAM では 27FFH からがアクセスされ、それぞれの V-RAM に書き込まれていたデータがこれから CPU によってアクセスするフォントデータを決定します。

どのフォントデータをアクセスするかは、前に選択された各テキスト V-RAM の最上位 1 バイトに書き込まれていたデータ計 3 バイトと、CPU からのアドレスによって決定されます。各テキスト V-RAM からのデータによって、CGROM, PCGRAM, 漢字 ROM の選択及びキャラクタの選択が行われ、CPU からのアドレスによって、そのキャラクタの何ライン目をアクセスするかを選択します。この選択のための規則は、「フォント表示」の項で説明したのとほぼ同様の規則にしたがいます。ただし、CRTC からのラスタアドレス (RA) の代わりに CPU からのアドレスがつかわれます。これについての詳細は後述します。

以上のようにして CGROM 等のアドレスが決定され、CGROM 等の準備が整った後に CPU のウェイト信号が解除され、CPU によってフォントデータの読み込み／書き込みが行われアクセスが終了します。実際には、CGROM 等のアドレス決定および準備期間として水平帰線期間に入ってから 5 キャラクタクロックの時間がとられます。

高速アクセスモードでのアクセスタイミングを下図に示します。

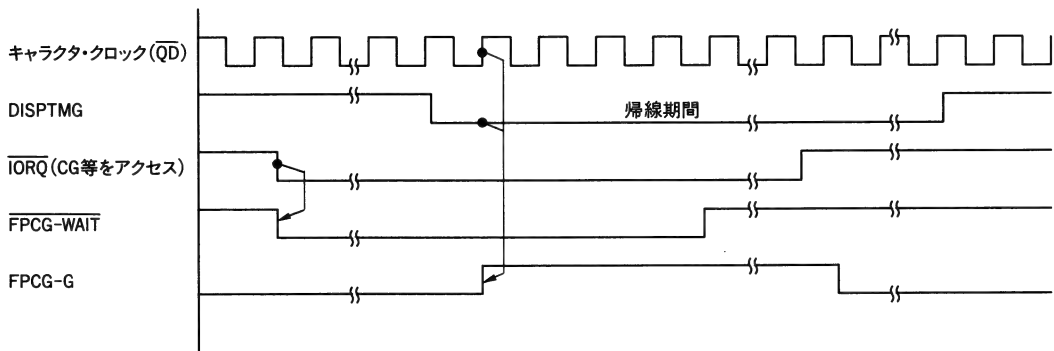


図4-22

以上のことから、高速アクセスモードでのアクセス・プログラム手順の概要は、以下のようになります。

(1) 各テキスト V-RAM の最上位 1 バイトずつ(テキスト V-RAM 37FFH, 漢字用テキスト V-RAM 3FFFH, アトリビュート V-RAM 27FFH), 及び画面管理用 I/O ポート等に、アクセスしようとするフォントを選択するためのデータを書き込む。

(2) 画面管理用 I/O ポート (I/O アドレス 1FD * H) のデータビット 5 (SPCG/FPCG 信号) を "1" に設定する。

(3) CGROM 等のマッピングされている I/O アドレスをアクセスする。それぞれの I/O アドレスは、次のようになっています。

CGROMI/O アドレス	14 * 0H ~ 14 * FH
PCGRAM・BLUEI/O アドレス	15 * 0H ~ 15 * FH
PCGRAM・REDI/O アドレス	16 * 0H ~ 16 * FH
PCGRAM・GREENI/O アドレス	17 * 0H ~ 17 * FH
漢字ROMI/O アドレス	14 * 0H ~ 14 * FH

このアドレスの下位4ビットが、テキスト V-RAM 等によって選択されたキャラクタフォントのうち、何ライン目をアクセスするかを選択するのに使用されます。

これで、ソフトウェア的な手段は終了し、あとはハードウェアが自動的にタイミングをとって実際に CGROM 等に対してアクセスを行い、1バイト分のデータアクセスが完了します。データが、複数バイトある場合には、この手順を必要なだけ繰り返します。

次に、フォントデータ選択のための規則について説明します。

フォントデータの選択は、各テキスト V-RAM からのデータと、CPU からのアドレスによって決定されます。

各テキスト V-RAM からのデータは、計3バイトで、CGROM・PCGRAM・漢字 ROM のうちどれを選択するか、及びそのなかでもどのキャラクタのデータをアクセスするかを選択するのにつかわれており、その規則は「フォント表示」のときとはほぼ同じです。相違点は、CGROM にアクセスするとき、8×8ドットフォントと8×16ドットフォントのどちらをアクセスするかを決定する規則で、表示のときは表示画面モードによって自動的に選択されていましたが、高速アクセスモードでは画面管理用 I/O ポート・ビット6 (CGSEL 8/16RA)の値によって選択されます。

詳細については図4-1・画面管理用 I/O ポートの内容を参照してください。

CPU からのアドレスは、下位4ビット (AB0~AB3)が、CRTC からのラスタアドレスに代わって、選択されたキャラクタのうちどのラインのデータをアクセスするかを決定するのにつかわれています。表示の場合、CGROM 等に与えられるアドレスの下位4ビットまたは3ビットをラスタアドレスが担当していましたが、その対応は表示画面モードによってまちまちでした。しかし、高速アクセスモードの場合、PCGRAM へのアクセスの一部を除いてつねに、CGROM 等へ与えられるアドレスの下位4ビットと、CPU からのアドレス下位4ビットの対応は、同じになっています。

以下に、CGROM、PCGRAM、漢字 ROM、それぞれにアクセスする場合に分けて説明します。

(1) CGROM

CGROM が選択されるのは、各テキスト V-RAM からのデータのうち、ROM/RAM 信号(アトリビュート VRAM・ビット5)と、CG/KANJI 信号(漢字用テキスト V-RAM・ビット7)がともに“0”のときです。

また、CGROM に与えられるアドレスは、画面管理用 I/O ポート・ビット6、各テキスト V-RAM からのデータ、CPU からのアドレス下位4ビットによって構成され、その内容は、以下のようになります。

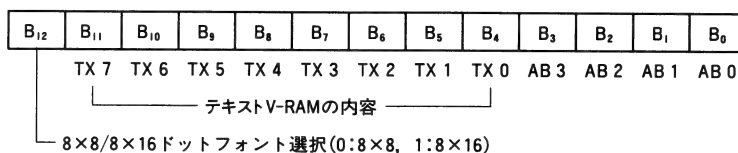


図4-23 CGROMに与えられるアドレスの内容

CGROM に書き込まれているデータは、大きく2つに分けられ、前半に8×8ドットフォントデータが、後半に8×16ドットフォントデータが書き込まれています。このうち、どちらを選択

するかは CGROM に与えられる最上位ビット (B12) によって決定されますが、ここには画面管理用 I/O ポート・ビット 6 (CGSEL 8/16RA 信号) の値がそのまま入ります。

画面管理用 I/O ポート (I/O アドレス 1FDG * H)

ビット 6 CGSEL 8/16RA (8/16 ラインフォント選択) 信号

0 : 8 ライン (8 × 8 ドット) フォント選択

1 : 16 ライン (8 × 16 ドット) フォント選択

また、8 × 8 ドットフォントは、2 バイトずつ同じデータが書き込まれているので、8 × 8 ドットフォントデータを順次アクセスする場合には CPU からのアドレスをダブルインクリメントする必要があります。

(2) PCGRAM

PCGRAM が選択されるのは、各テキスト V-RAM からのデータのうち、ROM/RAM 信号 (アトリビュート V-RAM・ビット 5) の値が、“1” のときです。

PCGRAM に与えられるアドレスは、各テキスト V-RAM からのデータと、CPU からのアドレス下位 4 ビットとで作成されますが、その構成によって PCG キャラクタ方式と PCG 外字方式の 2 つに分けられ、それぞれ PCG の 2 つの表示方式に対応しています。

2 つの方式のうち、どちらの方式でアクセスするかは、CG/KANJI 信号 (漢字用テキスト V-RAM・ビット 7) と、1/2 水準信号 (漢字テキスト V-RAM・ビット 4) の値によって選択されます。この 2 つのビットの値が、ともに “0” の時は PCG キャラクタ方式でアクセスが行われ、それ以降の場合には PCG 外字方式でアクセスが行われます。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
CG/ KANJI	L/R	ULINE	1/2 水準	ASCII 2			

CG/KANJI	1/2水準	テキスト表示
0	0	PCG キャラクタ方式
0	1	PCG 外字方式 (①)
1	*	PCG 外字方式 (②)

表4-19 漢字用テキスト V-RAM の内容と、PCG アクセス方式の選択

PCG キャラクタ方式は、1 キャラクタを 8 × 8 ドットで構成するもので、このときに PCGRAM に与えられるアドレスの内容は下図のようになります。

B ₁₀	B ₉	B ₈	B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀
TX 7	TX 6	TX 5	TX 4	TX 3	TX 2	TX 1	TX 0	AB 3	AB 2	AB 1
└────────── テキスト V-RAM の内容 ─────────┘										

図4-24 PCGRAM に与えられるアドレスの内容 (PCG キャラクタ方式)

図からもわかるように、CPU からのアドレスのうち、最下位の AB0 が使われずに AB1 から始まっています。したがって、偶数アドレスとそれに続く奇数アドレスでは同じラインの PCG を選択することになるので、PCGRAM を PCG キャラクタ方式で順次アクセスする場合には、CPU

からのアドレスをダブルインクリメントする必要があります。

PCG 外字方式は、8×8ドットのフォントを2つ縦に並べて、1キャラクタを8×16ドットで構成するもので、その構成が漢字フォントと同じになるためには漢字テキストと一緒に表示するとき等に使用すると便利な方式です。

PCG 外字方式のときに PCGRAM に与えられるアドレスの内容は、以下のようになります。

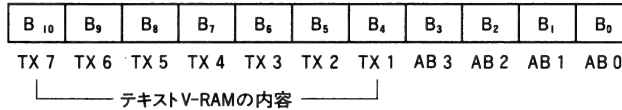


図4-25 PCGRAMに与えられるアドレスの内容(PCG外字方式)

CPU からのアドレスは、4ビットともすべて使われますが、テキスト V-RAM からのデータのうち最下位のビットが使われていません。したがって、偶数とそれに続く奇数のアスキーコードは、おなじ1つのキャラクタを選択します。この場合、このキャラクタを示すのにどちらのコードを使ってもかまいません。

また、アスキーコードの最下位ビットの代わりに、CPU からのアドレスの最上位ビットが入っているので、偶数のアスキーコードに対応する8×8ドットのフォントがライン0～7、続く奇数のアスキーコードに対応する8×8ドットのフォントがライン8～15のフォントデータになります。

(3) 漢字 ROM

各テキスト V-RAM からのデータのうち、ROM/RAM 信号(アトリビュート V-RAM・ビット5)が“0”で、CG/KANJI 信号(漢字用テキスト V-RAM・ビット7)が“1”のときには、アクセス対象として漢字 ROM が選択されます。

漢字 ROM は、第1水準・第2水準(第2水準はオプション)合わせて8個のROMに納められていますが、このうちのどのROMをアクセス対象にするかは、LEFT/RIGHT 信号(漢字用テキスト V-RAM・ビット6)、第1/第2水準選択信号(漢字用テキスト V-RAM・ビット4)、及び、漢字用テキスト V-RAM・ビット3の計3つのビットの状態によって選択されます。この内容は表4-15を参照してください。

このようにして選択された漢字 ROM に与えられるアドレスの内容は、以下のようになります。

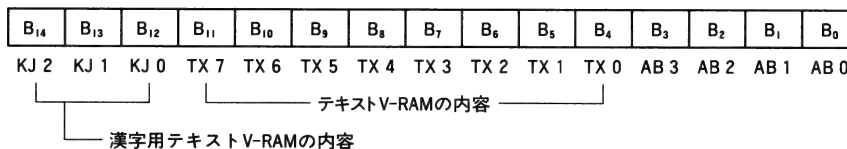


図4-26 漢字ROMに与えられるアドレスの内容

このとき、テキスト V-RAM と漢字用テキスト V-RAM に書き込まれる値は、区点コードや JIS 漢字コードとは、全く別のもので、この値が直接、漢字 ROM アドレスを指定するのに使われます。漢字 ROM アドレスを JIS 漢字コードから求める方法を以下に示します。

例えば漢字“右”(JIS 漢字コード 3126H)の漢字 ROM アドレスを求めると

- ・ JIS 漢字コードの上位バイトから 30H を引きます。
 $31\text{H} - 30\text{H} = 01\text{H}$
ただし JIS 漢字コードの上位バイトが 28H 以下の場合には 21H を引きます。
- ・ この値に 600H を掛けます。
 $01\text{H} \times 600\text{H} = 600\text{H}$
- ・ この値に 4000H を足します。
 $600\text{H} + 4000\text{H} = 4600\text{H}$
ただし JIS 漢字コードの上位バイトが 28H 以下の場合には 100H を足します。
こうして得られた値をテーブルデータといい、その行の最左列の漢字のフォントデータ収納アドレスを示しています。
- ・ さらに漢字 ROM アドレスを求めるには
漢字 ROM アドレス = テーブルデータ + (JIS 漢字コード下位バイト - 20H) × 10H
となります。
“右” の場合には
漢字 ROM アドレス = $4600\text{H} + (26\text{H} - 20\text{H}) \times 10\text{H} = 4660\text{H}$
となります。

次に JIS 漢字コードから漢字 ROM アドレスに変換するプログラムを示します。

リスト4-13 BIOS内ルーチンを利用しJIS漢字コードから漢字ROMアドレスへの変換

JISVRM EQU	2FB6HJISコード → 漢字ROMアドレス
KKVRM: LD	BC, 1A01H] BIOS ROMの状態を調べ、それをセーブする
IN	A, (C)	
AND	10H	
JR	Z, KKVR1	
LD	A, 1DH	
JR	KKVR2] BIOS ROMアクティブ
KKVR1: LD	A, 1EH	
KKVR2: PUSH	AF] DEに漢字コードをセット
LD	A, 1DH	
OUT	(00H), A	
LD	HL, KKVD T	
LD	E, (HL)	
INC	HL] DEに漢字コードをセット
LD	D, (HL)	
CALL	JISVRMBIOSコール
JR	NC, KKVR3	
XOR	A	
LD	E, A	
LD	D, A	
KKVR3: INC	HL] ROMアドレスをセーブ
LD	(HL), A	
INC	HL	
LD	(HL), E	
INC	HL	
LD	(HL), D	
POP	AF	
OUT	(00H), AROMを初期状態に
RET		
;		
KKVD T: DW	3021H	
DS	3	
;		
END		

また、X1turbo シリーズでは BIOS ROM の 2FB6H 番地 (JISVRM) に JIS 漢字コードから V-RAM のための漢字 ROM アドレスに変換するサブルーチンが用意されています。

X1 で漢字表示のためにグラフィック RAM に漢字パターンを展開したい場合など、漢字 ROM から直接フォントデータを読み出す必要があります。X1 では漢字 ROM からフォントデータを読み出す際には I/O ポートを利用します。表 4-20 は漢字 ROM の読み出しに関する I/O ポートの一覧表です。

I/Oポート	操 作 内 容	IN/OUT
0E80H	1. 収納アドレス下位データ設定	OUT
	2. 左側データ読み込みポート	IN
0E81H	1. 収納アドレス上位データ設定	OUT
	2. 右側データ読み込みポートと内部アドレスカウトアップ	IN
0E82H	1. [(0E82)H-(01)H]・チップセレクト ON	OUT
	2. [(0E82)H-(00)H]・チップセレクト OFF 増設用 EP-ROM セレクト	

(a) X1

I/Oポート	設定内容
1FD*Hの D ₅ ($\overline{\text{SPCG}}/\text{FPCG}$)	1: 高速アクセスモード
27FFHの D ₅ ($\overline{\text{ROM}}/\text{RAM}$)	0: ROM
3FFFHの D ₇ ($\overline{\text{CG}}/\text{KANJI}$) D ₆ ($\overline{\text{L}}/\text{R}$) D ₄ ($\overline{\text{T}}/2$ 水準) D ₃ ~D ₀	1: 漢字 0: 左フォント 1: 右フォント 0: 第1水準漢字 1: 第2水準漢字 漢字ROMアドレス上位 4 ビット
37FFH D ₇ ~D ₀	漢字ROMアドレス下位 8 ビット

(b) X1turbo

表4-20 漢字ROMのI/Oアドレス

次に漢字 ROM からフォントを読み出す手順を示します。

最初に JIS 漢字コード、または JIS 区点コードから前記の方法で漢字 ROM アドレスを求めます。前述の“右”の場合は漢字 ROM アドレスは 4660H となります。

- (1) I/O ポートの 0E80H に漢字 ROM アドレスの下位バイトを出力します。
- (2) I/O ポートの 0E81H に漢字 ROM アドレスの上位バイトを出力します。
- (3) I/O ポートの 0E82H に 01H を出力して漢字 ROM のチップセレクトを ON にします。
- (4) I/O ポートの 0E80H から左側のデータを読み込みます。
- (5) I/O ポートの 0E81H から右側のデータを読み込みます。このとき、漢字 ROM アドレスがハードウェアにより、自動的に 1 バイトカウントアップされます。
- (6) I/O ポートの 0E82H に 00H を出力して漢字 ROM のチップセレクトを OFF にします。

(3) から (6) までの動作を合計 16 回繰り返すことにより、漢字 1 文字のフォントデータの読み込みが完了します。また、(3) の動作から (4) の動作の間に 3 μ sec 以上の時間をおく必要があります。次に、フォント読みだしプログラムの例を示します。

リスト4-14 漢字ROMデータ読み込みプログラム例

```

KANACS: LD      BC, 1FD0H
        LD      A, 20H
        OUT     (C), A
        LD      BC, 27FFH
        XOR     A
        OUT     (C), A
        LD      BC, 3FFFH
        LD      HL, 401H .....*亜*の漢字アドレス
        LD      A, H
        OR      80H .....漢字ROM左フォント選択
        OUT     (C), A
        LD      BC, 3FFF
        LD      A, L .....漢字ROMアドレス下位8ビットの指定
        OUT     (C), A
        LD      BC, 1400H .....フォント読み出しI/Oポートアドレス
        LD      HL, DATA
        LD      E, 10H .....フォントライン数
GETKNL: IN      A, (C) .....フォントを読む
        LD      (HL), A .....作業域に書き込む
        INC     HL
        INC     C .....次のラインの指定
        DEC     E
        JR      NZ, GETKNL
        LD      BC, 3FFFH
        IN      A, (C)
        OR      40H
        OUT     (C), A
        LD      BC, 1400H
        LD      E, 10H
GETKNR: IN      A, (C)
        LD      (HL), A
        INC     HL
        INC     C
        INC     E
        JR      NZ, GETKNR
        RET
;
DATA:   DS      32
;
END

```

高速アクセスモードの設定

ROM選択

16回くり返す

読み出すフォントを右にする

漢字右フォントを16回読み込み、データをセーブする

4-4 特殊画面制御

X1はパレット機能、プライオリティ機能、スーパインポーズなどの特殊画面制御機能を持っています。

X1turboになって、アンダーライン表示機能、黒色制御機能が追加されました。さらにX1turboZでは、インターレース・スーパインポーズ、ページスクロール、ビデオデータの取り込み、量子化コントロール、モザイクコントロール、クロマキーコントロールなどの画像処理機能が拡張されました。

4-4-1 パレット機能

(1)X1のパレット機能

パレット機能は、グラフィック画面に出力される色を、グラフィック V-RAM の内容を書き換えることなく瞬時に別の色に変える機能です。

パレット機能はパレット回路により実現されています。パレット回路は3個のデータセクタ IC によって構成され、1個のデータセクタ IC は、内部に8ビットのデータを持っていて、グラフィック V-RAM からこのデータセクタに入力されるカラーコードによって、このうちの1ビットを画面へ出力しています。3個のデータセクタ IC は、それぞれ画面に出力される R・G・B の3つの信号に対応しており、したがって、1つのカラーコードに対して、各データセクタ IC から1ビットずつ計3ビットが、画面への R・G・B 信号として出力されます。

カラーコードと、それによって出力される各データセクタのビットは、以下のようになります。

カラーコード	S ₂ (G)	S ₁ (R)	S ₀ (B)	パレットコードのビット内容		
				G	R	B
0	0	0	0	(D ₀) ①	(D ₀) ②	(D ₀) ③
1	0	0	1	(D ₁) ①	(D ₁) ②	(D ₁) ③
2	0	1	0	(D ₂) ①	(D ₂) ②	(D ₂) ③
3	0	1	1	(D ₃) ①	(D ₃) ②	(D ₃) ③
4	1	0	0	(D ₄) ①	(D ₄) ②	(D ₄) ③
5	1	0	1	(D ₅) ①	(D ₅) ②	(D ₅) ③
6	1	1	0	(D ₆) ①	(D ₆) ②	(D ₆) ③
7	1	1	1	(D ₇) ①	(D ₇) ②	(D ₇) ③

表4-21 カラーコードとパレットコードのビット内容

一方、各データセクタの保持している8ビットのデータは、書き換えることができるので、結局各カラーコードに対して、画面へ出力される R・G・B 信号の状態を任意に変えることができます。

それぞれのデータセクタに値を設定するには、つぎの I/O アドレスを使います。

パレット設定用 I/O ポート

データセクタ IC①(BLUE).....I/O アドレス 10* *H
 データセクタ IC②(RED).....I/O アドレス 11* *H
 データセクタ IC③(GREEN).....I/O アドレス 12* *H

パレット用 I/O ポートに対するアクセスは出力のみ有効で、読み出すことはできないので注意して下さい。

以下にパレット設定プログラム例を示します。

リスト4-15 パレット設定プログラム例

```

PALSET EQU      1480H
PALET  EQU      07H
COLOR  EQU      04H
KSENGC:LD       A, 1DH      ] BIOS ROMをアクティブに
      OUT      (00), A
      LD       DE, PALET*256+COLOR ..... Dにパレットコード, Eにカラーコードをセット
      CALL    PALSET
      LD       A, 1EH      ] BIOS ROMをノンアクティブに
      OUT      (00), A
      RET
      ;
      END
  
```

なお、それぞれのデータセクタには、初期値として以下の値が設定されています。

データセクタ I C①(B L U E)..... A A H
データセクタ I C②(R E D)..... C C H
データセクタ I C③(G R E E N)..... F 0 H

(2)拡張パレット機能

(1) 拡張パレット機能の概要

X1turboZ では多色モードが新しく追加され、パレット機能も拡張されています。

グラフィックパレットは従来の機能の他に、パレットメモリとして内部パレットメモリと外部パレットメモリが追加されています。両者は多色モード時に使用され、表示モードによりその使い方が決まっています。

テキストパレットは次のような機能が追加されています。

コンパチモード..... 8 色中 8 色表示
多色モード..... 64 色中 8 色表示

内部パレットメモリ、外部パレットメモリは電源の立ち上げ時にハード的に初期化されます。ただし、この初期化は電源の立ち上げ時のみに行なわれ、前面の IPL SW を押しても初期化されません。

(2) グラフィックパレット

拡張グラフィックパレットとして、内部拡張グラフィックパレットメモリと外部拡張グラフィックパレットメモリがあります。内部拡張グラフィックパレットメモリは12ビット×8ワード、外部拡張グラフィックパレットメモリは12ビット×4096ワードのメモリ容量を持っています。いずれのパレットメモリも CPU からの読み出し／書き込みが可能です。

また、コンパチモードのグラフィックパレットメモリは turboZ 以前の機種と同じ手順でアクセスすることができます。

●グラフィックパレットのアクセス

多色モードのグラフィックパレットは内部と外部の2種類のパレットメモリがあります。内部パレットメモリは8ワード、外部パレットメモリは4096ワードあります。パレットメモリのアクセスは CPU からのアクセスと、CRT 表示データのアクセスの2種類があり、それぞれ以下のように対応しています。

パレットメモリー アドレス	11	10	9	8	7	6	5	4	3	2	1	0
CPUアクセス	AB4	AB5	AB6	AB7	AB0	AB1	AB2	AB3	DB4	DB5	DB6	DB7
CRT表示アクセス	QHC 3	QHC 2	QHC 1	QHC 0	QHB 3	QHB 2	QHB 1	QHB 0	QHA 3	QHA 2	QHA 1	QHA 0

表4-22 外部パレットメモリーアクセス (4096色の場合)

ここで、パレットメモリアドレスとは外部メモリの物理的なメモリアドレスを意味します。また CPU アクセスの時のアドレスは、アドレスバスの下位 8 ビットとデータバスの上位 4 ビットで構成されています。CPU のアクセス時のアドレスとデータの関係は次のようになります。

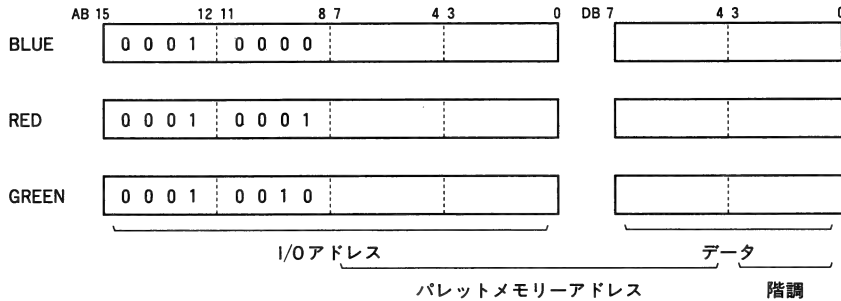


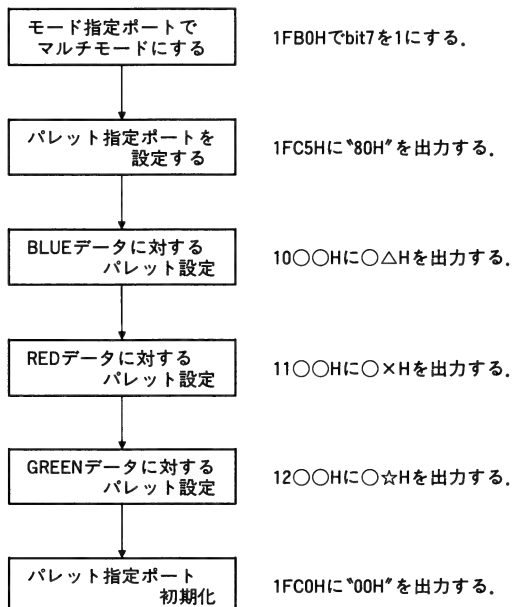
図4-27

パレットメモリのデータ(階調)を変更する場合は各色ごとに行う必要があります。BLUE, RED, GREEN についてそれぞれ4ビットの階調があり、各々16種の色を表示できます。これにより、最大4096色の組合せが可能となります。

CRT 表示アクセスの場合には、グラフィックデータに対応したメモリアドレスの内容が色調データとして出力されます。QHA0~3はBLUE, QHB0~3がRED, QHC0~3がGREENのデータとしてパレットメモリアドレスとなります。

拡張グラフィックパレットメモリのアクセスの一般の方法を次に示します。

(i) パレットメモリーに書き込む



この例は、オリジナルカラー ○ ○ ○ を △ × ☆ にする。
B R G

図4-28

(ii) パレットメモリーを読み込む

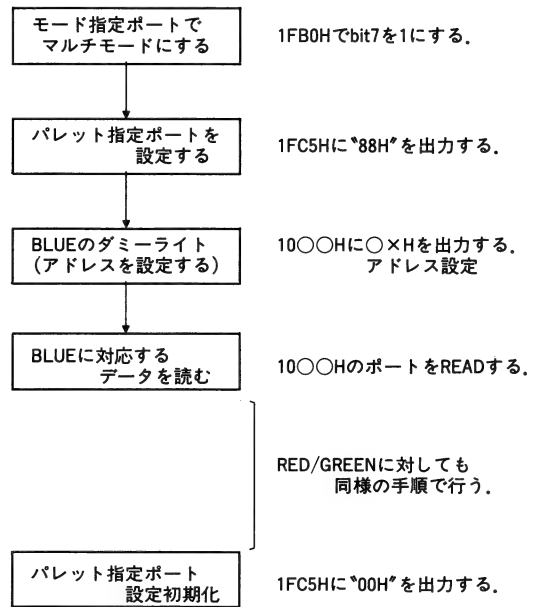


図4-29

●拡張パレットメモリのCPUアクセス

拡張パレットメモリはCPUから読み出し／書き込みすることができます。各表示モードにより表示できる色の数と、パレットメモリのアクセスできる領域(メモリアドレス構成)が異なります。

ここで、640×400モード時は、内部パレットメモリアドレスを使用するための8ワードの指定となります。この8ワードのメモリアドレスを選択するために、各色の先頭bit(BLUE:ビット0, RED:ビット4, GREEN:ビット8)が使用されます。他のビットは無意味です。

実際のCPUによるアクセスは読み出しと書き込みで若干異なっています。パレットメモリアドレスを設定する際、データ部の上位ビット(DB4~7)がアドレスの一部となっているため、パレットデータを書き込む時には、1回のOUT命令でB, R, Gいずれか1色のパレットデータを書き込むことができます。一方、読み出しの時にはまずOUT命令でアドレスを設定し、次にIN命令でパレットデータを読み込む必要があります。

パレットメモリにデータを書き込む例を示します。

- ・パレットメモリのF39H番地に、各色の階調を次のように設定するプログラム例です。

```
BLUE    3H
RED      AH
GREEN    7H
```

リスト4-16

```
BLPLT EQU    1000H
REDPLT EQU    1100H
GRPLT  EQU    1200H
PLTMOD EQU    1FB0H
PLTPT  EQU    1FC5H
SETPLT: LD     BC, PLTMOD
        LD     A, 80H
        OUT    (C), A
        LD     BC, PLTPT
        LD     A, 80H
        OUT    (C), A
        LD     BC, BLPLT
        LD     A, 03H
        OUT    (C), A
        LD     BC, REDPLT
        LD     A, 04H
        OUT    (C), A
        LD     BC, GRPLT
        LD     A, 05H
        OUT    (C), A
        RET
        END
```

多色モードにする
 パレットの書き込みモードにする
 ブルーに階調'3'を書く
 レッドに'4'
 グリーンに'5'

- ・パレットメモリの028H番地の値を読み出す例です。

リスト4-17

```
BLPLT EQU    1000H
REDPLT EQU    1100H
GRPLT  EQU    1200H
PLTMOD EQU    1FB0H
PLTPT  EQU    1FC5H
```

```

RDPLT: LD      BC, PLTMOD
        LD      A, 80H
        OUT     (C), A
        LD      BC, 1FC5H
        LD      A, 88H
        OUT     (C), A
        LD      (HL), DATA
        LD      BC, BLPLT
        LD      A, 01H
        OUT     (C), A
        IN      A, (C)
        LD      (HL), A
        INC     HL
        LD      BC, REDPLT
        LD      A, 01H
        OUT     (C), A
        IN      A, (C)
        LD      (HL), A
        INC     HL
        LD      BC, GRPLT
        LD      A, 01H
        OUT     (C), A
        IN      A, (C)
        LD      (HL), A
        INC     HL
        RET
;
DATA:   DS      3
;
        END

```

多色モードにする

パレットの読み込みモードにする

ブルーのパレットデータを読み、セーブする

レッドのパレットデータを読み、セーブする

グリーンのパレットデータを読み、セーブする

●パレットメモリの初期設定値

拡張パレットメモリは電源投入時にハードで自動的に設定を行っています。初期設定における外部パレットメモリおよび内部パレットメモリの値は次のようになっています。

パレット出力データ パレットメモリ アドレス			11	10	9	8	7	6	5	4	3	2	1	0
0H	0H	0H	0	0	0	0	0	0	0	0	0	0	0	0
0H	0H	1H	0	0	0	0	0	0	0	0	0	0	0	1
0	0	2	0	0	0	0	0	0	0	0	0	0	1	0
0	0	3	0	0	0	0	0	0	0	0	0	0	1	1
		⋮												
F	F	D	1	1	1	1	1	1	1	1	1	1	0	1
F	F	E	1	1	1	1	1	1	1	1	1	1	1	0
F	F	F	1	1	1	1	1	1	1	1	1	1	1	1

表4-23 外部パレットメモリ

パレット出力データ パレットメモリ アドレス			11	10	9	8	7	6	5	4	3	2	1	0
PA8 0	PA4 0	PA0 0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1	1
0	1	0	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表4-24 内部パレットメモリ

外部パレットメモリを初期化するプログラムは次のようになります。

リスト4-18 外部パレットメモリを初期化するプログラム

PLTMOD EQU	1FBOH	
PLTPT EQU	1FC5H	
INTPLT:LD	BC, PLTMOD	多色モードにする
LD	A, 80H	
OUT	(C), A	パレット書き込みモードにする
LD	BC, PLTPT	
LD	A, 80H	BC, Hでパレット指定 DEでパレットデータを指定
OUT	(C), A	
LD	DE, 0	
LD	BC, 1000H	
LD	H, 0	ブルーのパレットに書き込む
INTPL1:LD	B, 10H	
LD	A, E	
AND	0FH	
OR	H	レッドのパレットに書き込む
OUT	(C), A	
INC	B	
LD	A, E	
RRCA		
RRCA		
RRCA		グリーンのパレットに書き込む
RRCA		
AND	0FH	
OR	H	
OUT	(C), A	4096回くり返す
INC	B	
LD	A, D	
AND	0FH	
OR	H	
OUT	(C), A	
LD	A, H	
ADD	A, 10H	
LD	H, A	
JR	NC, LOOP2	
INC	C	
LOOP2: INC	DE	
LD	A, D	
AND	0FH	
JR	Z, LOOP1	
RET		
END		

●グラフィック RAM データとパレットの関係

表示データはグラフィックデータからパレット機能部に入力されます。入力されるデータとして、QHA0~3, QHB0~3, QHC0~3の3系統あり、それぞれBLUE, RED, GREENに対応しています。それぞれのデータは各モードにより、その有効、無効のビットが決っています。その関係を下に示します。

	QHC 0 1 2 3	QHB 0 1 2 3	QHA 0 1 2 3
640×400	* — — —	* — — —	* — — —
640×200	* * — —	* * — —	* * — —
320×400	* * — —	* * — —	* * — —
320×200	* * * *	* * * *	* * * *
2P P=0 320×200	* * — —	* * — —	* * — —
2P P=1 320×200	— — * *	— — * *	— — * *

*有効
—無効

表4-25

640×400の時は、内部パレットメモリを使用し入力されるデータ(QHA0, QHB0, QHC0)がメモリアドレスとなり、8色表示ができます。

640×400以外のモードはすべて外部パレットメモリを使用します。無効ビットに関しては、有効ビットを拡張してメモリアドレスを作り、外部パレットメモリをアクセスします。

(3) テキストパレット

テキストパレットは、テキストデータCEA, CEB, CEC(各々BLUE, RED, GREENに対応)から8通りのテキストパレットアドレスを発生します。

テキストパレットは8ワード×6ビットのメモリ構成になっており、アドレスを指定することにより、B, R, G各色に対して2ビット、計6ビットのデータを出力します。さらにこの2ビットを拡張して各色4ビットにして、合計12ビットを出力します。

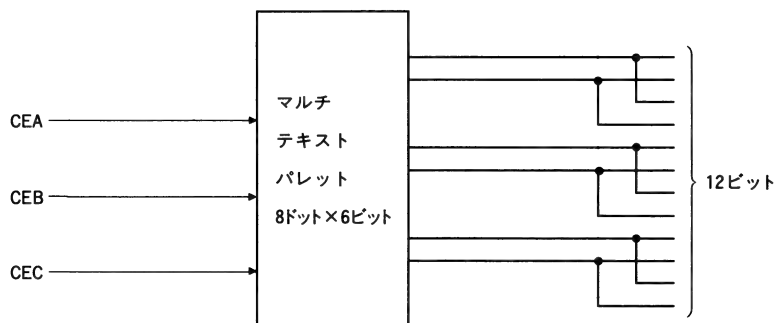


図4-30 テキストパレット機能の概念図

●テキストパレットのCPUアクセス

テキストパレットのI/Oアドレスは1FB8H～1FBFH となり、パレットメモリのアドレスと1：1に対応しています。

I/O アドレスとテキストパレットメモリアドレスの対応は次のようになります。

メモリアドレス	0	1	2	3	4	5	6	7
I/Oアドレス	1FB8	1FB9	1FBA	1FBB	1FBC	1FBD	1FBE	1FBF

表4-26

※ただし、1FB8H 番地についてはアクセスすることができません。1FB8H 番地は常に“0”に設定されています。

パレットメモリデータとCPU 書き込みデータの関係は次のようになっています。

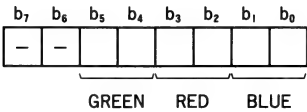


図4-31

このようにして書き込まれたデータは、各色ごとに拡張されて、以下のように12ビットの色調データを作り出します。

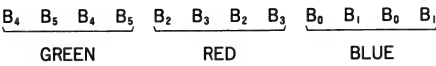


図4-32

テキストパレットをアクセスするには、通常のCPU のI/O 命令を実行します。なお、このパレットはCPU から読み出し／書き込みが可能です。

例

```
LD  BC, 1FB0H
LD  A, 80H
OUT (C), A
LD  BC, 1FB9H
LD  A, 31H
OUT (C), A
```

多色モードに設定。

テキストパレットメモリにデータ書き込む

●テキストパレットの初期化

テキストパレットは電源の立ち上げ時、ハードにより初期化されます。初期化の値は次のようになっています。

	11	10	9	8	7	6	5	4	3	2	1	0
	GREEN				RED				BLUE			
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	1	1	1
0	1	0	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	1	1	1	1	1
1	0	0	1	1	1	1	0	0	0	0	0	0
1	0	1	1	1	1	1	0	0	0	0	1	1
1	1	0	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1

表4-27 テキストパレットの初期値

4-4-2 アンダーライン表示機能

アンダーラインは、表示画面が低解像度の40(80)×20行、40(80)×10行モード、及び、高解像度の40(80)×20行モードのときに表示が可能です、1文字単位に指定することができます。

表示画面モードが25行モードおよび、12行モードのときにCRTCを設定しなおして、1画面あたりの行数を減らし、その分1行あたりのラスタ数を以下のように増やしたものが、20行モード、10行モードです。

	標準モード		アンダーライン表示モード	
	40(80)×25行	40(80)×12行	40(80)×20行	40(80)×10行
低解像度	8ラスタ/行	16ラスタ/行	10ラスタ/行	20ラスタ/行
高解像度	16ラスタ/行	32ラスタ/行	20ラスタ/行	

表4-28

アンダーライン表示モードでは、この増えたラスタ部分をつかってアンダーラインを表示します。

次に、アンダーラインを表示させる手順とそれぞれの過程での表示例を示します。

(1) 漢字の場合

- ① CRTC等を設定して、アンダーライン表示モードにします。これによって1行あたりのラスタ数が増えますが、その部分にはキャラクタの先頭部分が繰り返し表示されます。
- ② 25(12)/20(10)行(画面管理用I/Oポート(1FD * H)ビット7)を“1”に設定します。これで、増えたラスタ部分のテキスト表示と、グラフィック画面の表示が停止され、アンダーライン表示用の空白部が確保されます。
- ③ アンダーラインを表示させたい位置に対応する漢字テキスト用V-RAMのビット5(UNDER LINE信号)を“1”に設定します。これでアンダーラインが表示されます。

(2) テキストの場合

- ① CRTC等を設定して20行モードにします。1行のラスタ数は、8ラスタから2ラスタ増えて10ラスタ表示となります。増えた2ラスタには、キャラクタの先頭の部分がくり返し表示されます。
- ② 25/20行切り換えビットをアクティブにします。これによって、増えた2ラスタの部分の表示がカットされ、行間にスペースができます。

- ③キャラクタ“A”のアンダーライン・ビットをアクティブにします。これによって、キャラクタ“A”の9ラスタ目にアンダーラインが表示されます。

また、アンダーラインデータは、表示が停止されているグラフィックデータの代わりに画面に表示されます。したがって、パレット回路を利用して表示色を変えることができます。実際には、アンダーラインデータは、グラフィックデータのBLUEとしてパレット回路に出力されています。よって、パレットレジスタを変更してBLUEを他の色に変えることによって、アンダーラインの表示色を変えることができます。

また、アンダーライン表示機能は、ディスプレイに対してのみ有効な機能で、プリンター等には出力されません。

4-4-3 プライオリティ機能

X1シリーズの表示画面は、テキスト画面とグラフィック画面からなっています。通常は、テキスト画面はグラフィック画面より優先的に表示されますが、プライオリティ機能は、この優先順位を変えることができる機能で、グラフィックの任意のカラー(複数)をテキスト画面に対して優先的に表示させることが出来ます。

プライオリティ機能の原理を説明します。

プライオリティ回路は、データセクタICとマルチプレクサICで構成されています。

データセクタICは、グラフィックV-RAMからのカラーコードによって、内部にラッチされている8ビットのデータのうちから1ビットのデータをマルチプレクサに出力しています。

マルチプレクサICは、データセクタから出力された1ビットのデータが“1”のときには、グラフィック画面のデータを、“0”のときには、テキスト画面のデータを画面に出力します。また、テキスト画面のデータがない場合には、グラフィック画面のデータを出力します。

したがって、データセクタICにラッチされているデータのうち、カラーコードに対応したビットを“1”に設定することで、そのカラーコードを持ったグラフィックデータをテキスト画面

場合	S ₂ (G)	S ₁ (R)	S ₀ (B)	Y ₀	ビット 内 容	動 作
0	0	0	0	D ₀	0	テキストは、バック色より優先する。
1					1	テキストは、バック色と同じになる。
2	0	0	1	D ₁	0	テキストは、青色より優先する。
3					1	青色はテキストより優先する。
4	0	1	0	D ₂	0	テキストは、赤より優先する。
5					1	赤は、テキストより優先する。
6	0	1	1	D ₃	0	テキストは、マゼンタより優先する。
7					1	マゼンタは、テキストより優先する。
8	1	0	0	D ₄	0	テキストは、緑より優先する。
9					1	緑は、テキストより優先する。
A	1	0	1	D ₅	0	テキストは、シアンより優先する。
B					1	シアンは、テキストより優先する。
C	1	1	0	D ₆	0	テキストは、黄色より優先する。
D					1	黄色は、テキストより優先する。
E	1	1	1	D ₇	0	テキストは、白色より優先する。
F					1	白色は、テキストより優先する。

表4-29 プライオリティの組合せ

より優先して表示させることができます。

以下に、データセレクトにラッチされたデータと優先順位の関係を示します。

データセレクトには、初期状態として 00H、つまり、テキストがすべてのグラフィックカラーコードに優先して表示されるデータが設定されています。

優先順位の変更はシステム I/O ポートのプライオリティ設定用 I/O ポート (13 * * H) を介して行います。例えば、グラフィック画面の青、マゼンダ、シアン、黄をテキスト画面より優先して表示したい場合、プライオリティ設定用 I/O ポートに、(01101010) 2 = 6AH を出力します。

4-4-4 黒色制御機能

(1) 概要

従来の X1 では、テキスト画面では黒色の表示は不可能でした。

グラフィック画面では、プライオリティ機能で適当な色をテキスト画面より優先順位を高めて、パレットによって透明にすることによって、テキスト画面に対する黒色表示を可能にしていました。しかし、スーパーインポーズ時にはこの部分にも TV 画面が写ってしまい、黒色は表示できませんでした。

一方、X1turbo, turboZ では、黒色制御機能を使うことにより、テキスト画面では、グラフィック画面と TV 画面に対する黒色を、グラフィック画面では、テキスト画面と TV 画面に対する黒色を表示することが可能になりました。

(2) X1turbo の黒色制御機能

テキスト画面では、カラーコード 0 (透明) 以外の 7 色のうちの任意の 1 色を指定して黒にすることができます。黒色制御回路は、指定された色とテキスト画面信号の色を比較し、一致した場合には、画面への R・G・B 出力信号を "0" にするとともに、スーパーインポーズからの画面出力信号もカットします。これにより、画面、スーパーインポーズ出力ともに黒を表示することが可能になりました。

グラフィック画面では、カラーコード 0 (透明) とカラーコード 1 (青) に限って、TV 画面に対して黒を表示することができます。

また、黒色変換を使用して、コンピュータ画面のブランキング期間を TV 画面に対して黒にすることもできます。これは、バック色が透明の時のみ可能です。

ただし、X1turbo 専用ディスプレイ以外では、テキスト画面に対する黒色表示はできませんが、TV 画面を隠すことはできません。また、コンピュータのブランキング期間を黒色にすることもできません。

データ内容	コントロール内容	備 考
DB ₀	テキスト B	これらのデータで、黒変換する色を指定します。
DB ₁	テキスト R	
DB ₂	テキスト G	
DB ₃	テキスト 黒	このビットが H で指定色を黒変換します。
DB ₄	グラフィック 透明→黒変換	
DB ₅	グラフィック 青→黒変換	
DB ₆	コンピュータブランキング期間→黒変換	

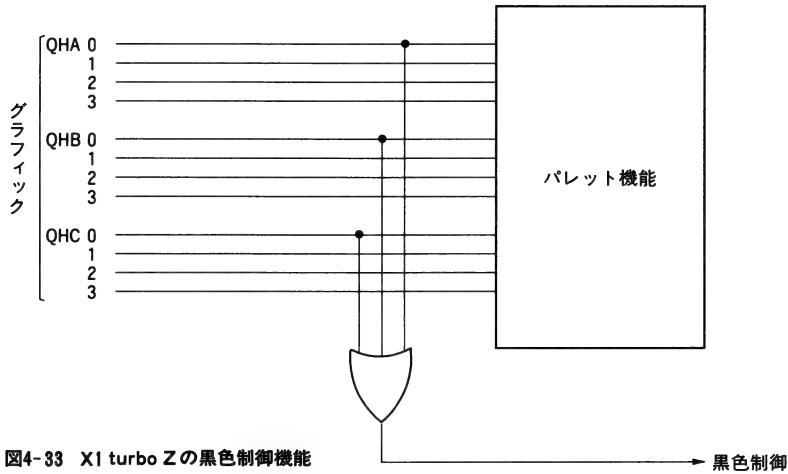
表4-30 黒色制御I/Oポート (I/OアドレスIFE*H)の内容

黒色制御機能は、黒色制御 I/O ポート (I/O アドレス 1FE * H) によってコントロールされます。各ビットの意味は以下になっています。

(3) X1turboZ の黒色制御機能

黒色変換制御の原理はコンパチモードと多色モードで変更されておりません。また、機能自身も変わっていません。ただし、turboZ では色表示が今までの 8 色から最大4096色表示まで拡張されているために、黒色制御機能の内容が変わっています。これは、グラフィックのデータが各 1 ビットデータから 4 ビットデータになったためです。

この概念図を下に示します。



つまり各色の最上位ビットにより黒色制御を行っているための階調が低いとき完全な透明色でない場合も、黒色制御機能が動作します。各表示モードと有効ビットの関係により、黒色制御が動作する範囲が決まります。

	QHA				QHB				QHC				黒色制御が動作する種類
	0	1	2	3	0	1	2	3	0	1	2	3	
640×400	*	—	—	—	*	—	—	—	*	—	—	—	1
640×200	*	*	—	—	*	*	—	—	*	*	—	—	8
320×400	*	*	—	—	*	*	—	—	*	*	—	—	8
2P=OFF 320×200	*	*	*	*	*	*	*	*	*	*	*	*	512
2P=ON page=0 320×200	*	*	L	L	*	*	L	L	*	*	L	L	8
page=1 320×200	L	L	*	*	L	L	*	*	L	L	*	*	64

表4-31

たとえば、320×200モードで2画面モードの page1 は、黒色制御を ON にすると画面がすべて黒になります。

なお、ソフト的なコントロールはコンパチモードと同じ方法で行うことができます。

4-4-5 スーパーインポーズ機能

(1) 概要

X1シリーズには、コンピュータ画面とTV等の画面を合成するスーパーインポーズ機能が装備されています。

X1では、画面の合成を2つの過程に分けて考えています。

1つ目の過程は、TV画面信号と、コンピュータの画面信号との同期をとる過程で、これで、専用ディスプレイ上での合成ができます。しかし、ここまでは、あくまでも専用ディスプレイ上での画面合成で、合成された画面を専用以外のディスプレイに表示させたり、VTR等に記録したりすることはできません。

次の過程では、実際にTV等の映像信号と、コンピュータの画像との合成画面、あるいはコンピュータのみの画像をNTSC信号に変換します。これで、合成画面がビデオ信号として出力されるので、合成された画面を専用以外のディスプレイに表示させたり、VTR等に記録したりすることができます。これには、専用のスーパーインポーズ回路を使います。

(2) 自動同期制御(ASC)回路

CRT画面で、一般放送やVTR、VHD、TVカメラ等のビデオ画像と、コンピュータ画像を合成するには、TV信号・ビデオ信号の水平・垂直同期信号に同期して、コンピュータの画像信号を送出しなければなりません。X1では、この過程を自動同期信号(ASC)回路によって行っています。

X1では、表示タイミング等をCRTCで作成しており、画像構成、水平・垂直同期信号、表示時間等は、画像表示の最小単位(1ドット)の周波数をCRTCで分周することで作られます。

そこで、まずCRTCで作られる同期信号の周期を、正規のNTSC規格の同期信号の周期より若干短めに設定しておきます。

スーパーインポーズにはいると、ASC回路内のデジタル・フェーズ・ディテクタで、CRTCからの同期信号と、外部からの同期信号との位相差を検出し、その差分の時間(MIX)、CRTCのクロック入力を停止しておき、CRTC側の同期の不足分を補うかたちで位相、周波数を合わせこんでいきます。

この補正動作は、同期信号の各周期ごとにおこなわれます。

ただし、高解像度モードのときは、水平同期信号の周期がNTSC規格のそれと大幅に違うため、スーパーインポーズ動作をおこなうことはできません。

自動同期制御(ASC)回路は、スーパーインポーズモードにはいると自動的に動作を開始します。スーパーインポーズの設定は、サブCPUにコマンドを送ることでおこないます。例を以下に示します。

(3) スーパーインポーズ回路

自動同期制御回路でつくられたコンピュータの画像信号は、外部の映像信号と同期がとれているだけで、画像信号はR・G・B信号のままです。したがって、専用ディスプレイ上では、画面の合成ができますが、外部に合成画面として取り出すことはできません。

合成画面を外部に出力するには、この合成画面をNTSC信号に変換してから出力しなければなりません。これを行うのが、スーパーインポーズ回路です。

自動同期制御回路では、同期信号に、コンピュータの画像信号を同期させ、この同期させたコンピュータのR・G・B信号をスーパーインポーズ回路に送ります。

スーパーインポーズ回路に入力されたR・G・B信号から、輝度信号(Y)と色差信号(C)がつかわれ、あらかじめ分離させておいたビデオ信号の輝度信号、色差信号とそれぞれ混合され、ふたたび NTSC 規格の信号に合成されてから、ビデオ信号出力信号端子に出力されます。

このとき出力される信号は、NTSC 規格の信号になりますので、ビデオ信号入力端子をもったディスプレイやモニター上に、合成画面を表示させることができます。また、ビデオ等に合成画面を記録しておくこともできます。

(4) インターレース・スーパーインポーズ(turboZ)

・概要

X1turbo では、スーパーインポーズを行う場合に高解像度ではその機能を使用することはできませんでした。このため25行または20行表示の文字は8×8ドットの文字フォントを使用していました。このため、文字品位は16×8フォントの場合よりも劣るようになっていました。これを解決したのが、インターレース・スーパーインポーズ機能です。

これは、モニターのインターレース機能を利用して、みかけのライン数を2倍にする機能です。これにより、25行/20行表示の場合も16×8フォントを使用することができるようになりました。表示文字数とフォントの関係は下の表のようになります。

	80×25	80×20	80×12	80×10	40×25	40×20	40×12	40×10	(キャラクタ)
通常のスーパーインポーズ	8×8	8×8	16×8	16×8	8×8	8×8	16×8	16×8	} (ドット)
インターレーススーパーインポーズ	16×8	16×8	16×8	16×8	16×8	16×8	16×8	16×8	

表4-32

・設定方法

インターレース・スーパーインポーズ機能を利用する場合には、通常のスーパーインポーズの状態、HIRES Si ビット ON にします。

HIRES Si はモード指定ポート(1FB0H)のビット0に対応しています。

なお、インターレース・スーパーインポーズを行う場合には次のような点に注意してください。

1. turboZ 専用モニター を使用してください。
2. 裏面の VideoIN にジャックを入れて前面パネルの VTR RECORD を ON にしたままで、ビデオ信号を入力しないという状態にはしないでください。

4-4-6 スクロール機能

(1) X1turbo のスクロール機能

・概要

X1 は、スーパーインポーズ時の特殊な画面制御機能として、スクロール機能をもっています。スクロール機能は、スーパーインポーズ時にコンピュータ画面のみを上下方向にスクロールさせる機能です。

・設定方法

スクロール機能を使用するには、まず8255②ポートC(コントロールレジスタI/Oアドレス1A03H)のビット4(スクロールON/OFF信号)を"0"にします。さらに、CRTCのレジスタ5(I/Oアドレス1800H, 1801H)を次の表に示した値に設定すると、スクロールします。

レジスタの値(10進数)	0 ← 2 ← 4 ← 6	8 → 10 → 12 → → 30
スクロール方向	上方向	下方向
速度	速い ←	→ 速い

表4-33 CRTC・レジスタ5の値とスクロールの関係

CRTC・レジスタ5には、0～31の値を設定することができますが、スクロール設定値としてはからなず偶数を指定してください。

また、スクロールを止めて画面をもとに戻すにはまず、8255②ポートC(コントロールレジスタ I/O アドレス 1A03H)のビット4(スクロール ON/OFF 信号)を“1”にします。次に、CRTC・レジスタ5(I/O アドレス 1800H, 1801H)を、このレジスタに設定されていたもとの値に戻します。

以下に、スクロール設定プログラム例を示します。

リスト4-19 スムーズ・スクロール設定プログラム例

```
PIOADD EQU 1A03H
CRTCAD EQU 1800H
SSCRL: LD BC, CRTCAD
      LD A, 05H
      OUT (C), A
      LD A, 04H
      INC BC
      OUT (C), A
      LD A, 08H
      LD BC, PIOADD
      OUT (C), A
      RET
      ;
      END
```

CRTC Reg #5に4を書き込む(上にスクロール)

8255②ポートCのビット4をリセットする

(2) X1turboZ のスクロール

・概要

X1turboZ スクロールには従来のスクロール動作の他に、1ページ(1画面)分のみスクロールをする1ページスクロールモードがあります。

スクロールには、スクロールアウト、スクロールインアウトの2つの機能があります。

スクロールアウトとはコンピュータ画面を表示している状態から上または下へ消えていくモードです。

またスクロールインアウトとは、コンピュータ画面が消えた状態からコンピュータ画面が上または下から表示を開始し、全画面を表示してからそのまま動いて消えていくモードです。

また、スクロール以外に画面全体を一度に消す機能も持ってます。

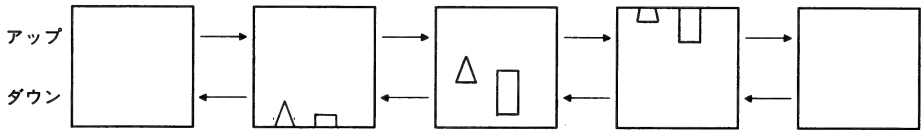


図4-34 スクロールイン・アウトモード

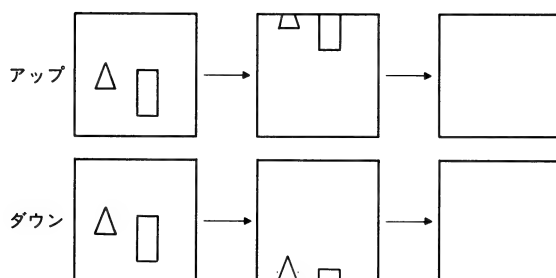


図4-35 スクロールアウト

・設定方法

次にスクロール指定ポートの説明をします。

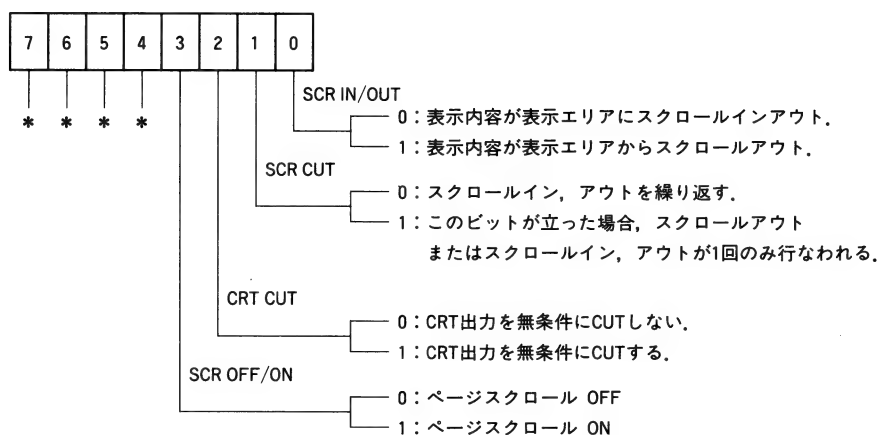


図4-36 スクロール指定ポート 1FC4H

このポートはコンパチ／多色いずれのモードでも有効です。D3が“0”のとき、D2-D0ビットは無効となります。

スクロール停止モードに設定してからページスクロール指定ポート(1F4CH)に各モードに応じたデータを設定し、上または下へのスクロール命令を出すことによりページスクロールを行うことができます。

スクロールイン・アウトモードの設定方法

上方向へのスクロール

SCROLL	0	スクロール停止
OUT	1FC4H, 0AH	スクロールポート設定
SCROLL	X	スクロールスタート
		X: スクロールスピード1～3

下方向へのスクロール

SCROLL	0
OUT	1FC4H, 0AH
SCROLL	-X

スクロールアウトモードの設定方法

上方向へのスクロール

SCROLL 0

OUT 1 FC 4 H, 0 BH

SCROLL X

スクロール停止

スクロール指定ポート設定

スクロールスタート

X: スクロールスピード 1 ~ 3

下方向へのスクロール

SCROLL 0

OUT 1 FC 4 H, 0 BH

SCROLL -X

4-4-7 画像処理

(1) 概要

turboZ で追加された機能として、リアルタイムビデオデータの取り込み機能があります。なお、取り込み機能は低解像度時のみ有効です。

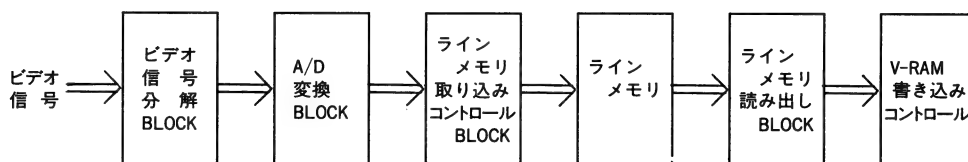


図4-37

ビデオ信号分解 BLOCK:

テレビモニタ等の合成されたビデオ信号を B, R, G の 3 つの信号に分解します。

A/D 変換 BLOCK:

B, R, G の各アナログ信号を A/D 変換して、各々 4 ビットのデジタル信号にする BLOCK です。

ラインメモリ取り込みコントロール BLOCK:

この BLOCK では A/D 変換されたデータをラインメモリに書くために、メモリのコントロールを行うと同時に、入力されたデータの量子化および縦方向/横方向のモザイク化を行っています。

モザイクの大きさおよび量子化にはソフトコントロールが必要です。

ラインメモリ:

ラインメモリは 910 ワード分のデータ容量を持っている FIFO メモリです。

ラインメモリの読み出し BLOCK:

ラインメモリは FIFO の構成になっており、書き込みと読み出しは個々にコントロールすることができます。テレビモニター等のビデオデータを取り込んで、V-RAM に書き込む場合、ライン単位でずれが発生するため、読み出し BLOCK でそのずれ補正を行っています。

V-RAM 書き込みコントロール:

V-RAM は通常のグラフィックデータと共有しているため、取り込みデータを V-RAM に書く

ためのコントロールが必要となります。ここでは、反転取り込み、クロマキーコントロール等を行っています。

取り込みコントロールをするためには、まずモードを多色モードにした後、表示モードを低解像度にして、その後取り込みイネーブルにする必要があります。

取り込みに関係して、ソフトでコントロールする必要のある部分は次の部分です。

- ・モザイクの大きさの指定
- ・量子化のビットコントロール
- ・取り込み位置の補正值の設定
- ・クロマキーコントロール
- ・反転取り込み

(2) 取り込み開始コントロール

通常取り込みを行う場合には、以下のポートにコントロールビットをセットする必要があります。

1) モード指定ポート(1FB0H)

ビット3： 1・・・取り込み ON bit

ビット7： 1・・・多色モード

2) スーパーインポーズ

80C49 へ E7H, 1FH のコードを送信し、スーパーインポーズとします。

(E7H, 1EH)

3) 画面管理ポート(1FD0H)

bit 0： 0・・・低解像度にする(200ライン)。

以上の操作で取り込みは実行できますが、画面管理ポートをコントロールする場合には他の機能(表示機能)にも影響しますので注意して下さい。

(3) 量子化コントロール

量子化は、A/D 変換された各色 4 ビットデータに対して量子化を行うコントロールブロックで、4 種の量子化コントロールできます。この選択はプログラムで行います。4 種類の量子化機能について入力データと、出力データの関係は次のようになっています。

入力データ	4 ビット量子化	3 ビット量子化	2 ビット量子化	1 ビット量子化
D 3	D 3	D 1	D 1	D 0
D 2	D 2	D 2	D 0	D 0
D 1	D 1	D 1	D 1	D 0
D 0	D 0	D 0	D 0	D 0

各量子化によって表示色は次のようになります。

- 4 ビット量子化 4096色
- 3 ビット量子化 512色
- 2 ビット量子化 64色
- 1 ビット量子化 8色

量子化コントロールは、1FC2H ポートの bit6, 7 によって選択することができます。このポートは、モザイクコントロールと同じポートになっています。コントロールポートの内容については(4)モザイクコントロールで説明します。

なお、このポートは多色モードに設定されていればCPUで読み出し／書き込みが可能です。必要なデータビットをONにしてI/O命令で出力するだけで機能をコントロールすることができます。

(4) モザイクコントロール

モザイクの大きさのコントロールはX方向、Y方向それぞれ独自にその大きさを指定することにより行います。

- ・横方向のモザイク

入力ドットクロックをコントロールすることにより行っています。

- ・縦方向のモザイク

水平帰線信号(CHsync)により、ラインメモリのライトイネーブル信号をコントロールすることによって行っています。

- ・量子化およびモザイク化は、1FC2H ポートを次の表のようにコントロールすることにより行うことができます。

- ・量子化で b6, b7 が 0 の時は、4 ビット量子化となります。

量子化	Yモザイク	Xモザイク	
D ₇ D ₆	D ₅ D ₄ D ₃	D ₂ D ₁ D ₀	
0 0	0 0 0	0 0 0	通常取り込み
	0 0 1	0 0 1	2ドットモザイク
	0 1 0	0 1 0	4ドットモザイク
	0 1 1	0 1 1	8ドットモザイク
	1 0 0	1 0 0	16ドットモザイク
	1 0 1	1 0 1	32ドットモザイク
		1 1 0	64ドットモザイク
0 1			3bit階調
1 0			2bit階調
1 1			1bit階調

表4-34

(5) 画像取り込み位置補正

A/Dコンバータにより出力されるR,G,B各4ビットのデータは、モニター表示用ドットクロックに同期してデジタル化されます。ビデオデータがA/D変換されて、各ブロックを通り、表示ブロックに到達するまでに、相当のディレイが発生します。このため取り込みデータをモニターにそのまま表示しようとする、どうしてもズレた表示になります。このため、このズレを補正するために1ライン分のメモリを置き、このズレを補正しようとするのが位置補正です。ただし、このラインメモリを置くことにより、実際の表示は1ライン下方に移動します。

ズレ補正は、ズレ補正ポートに補正值を設定することにより行うことができます。

ズレ補正ポートは1FC1Hで、補正值として0H~FFHまでの任意の値が設定できます。

turboZでは、水平方向のズレ補正值を以下のように設定すると正しく位置が補正できます。

40文字モードの時・・・・・・40(10進)

80文字モードの時・・・・・・48(10進)

ズレ補正が正しく設定されていないと、画面が左右にズレて表示されます。
ズレ補正の設定はI/O 命令により行います。

プログラム例(80文字の時)

LD BC, 1FC1H

LD A, 48

ポート 1FC1H に48を出力する.

OUT (C), A

(6) クロマキーコントロール

クロマキー機能は指定された色を透明色にする機能です。

クロマキー指定ポートは以下のようになっています。

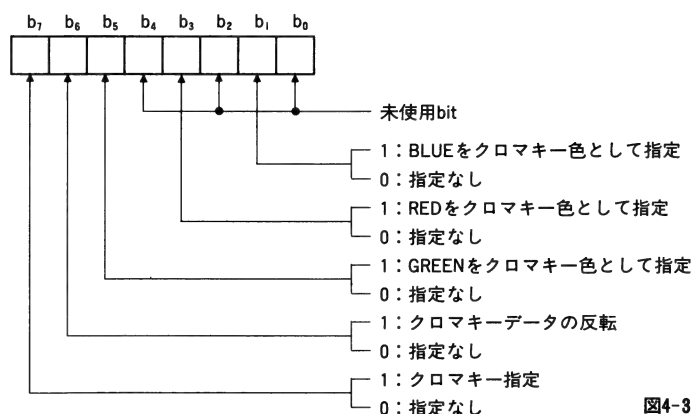


图4-38 &H1FC3

ビット7が0の時は、ビット6～ビット0はすべて無効となります。つまり、ビット7がクロマキー機能のマスターイネーブルの役目となります。ビット5, 3, 1は各色のイネーブルビットとなります。また、ビット6はクロマキーデータの反転を行うようになっています。

クロマキー機能は、各色の bit0 について作用し、他のビットについては関係がありません。

(7) 反転機能

取り込みデータの反転機能は、取り込んだデータの bit の反転を行います。反転機能はラインメモリから読み出して、V-RAM に書く直前で行います。反転を行った場合と行わなかった場合は、それぞれ次のようになります。

入力データ	反 転 出 力	非反転出力
D_0	$\overline{D_0}$	D_0
D_1	$\overline{D_1}$	D_1
D_2	$\overline{D_2}$	D_2
D_3	$\overline{D_3}$	D_3

表4-35

反転するか否かは、モード指定ポート(1FB0H)のビット2によってコントロールされます。

モード指定ポート (1FB0H) の内容

0 取り込み反転しない。

1 取り込み反転する.

第 5 章

サブCPU

5-1 サブ CPU

5-1-1 サブ CPU の構成

X1 シリーズでは、メイン CPU の負担を軽減するために、2 つのワンチップ CPU を搭載しています。これらの CPU は、キーボード、カセットなどのコントロールを行い、メイン CPU はサブ CPU に命令を送るだけで、各種の処理が行われるようになっています。

サブ CPU とメイン CPU 周りの構成を、図 5-1 に示します。但し、この図は X1turbo model -20 の場合で、機種によって構成は若干の違いがあります。

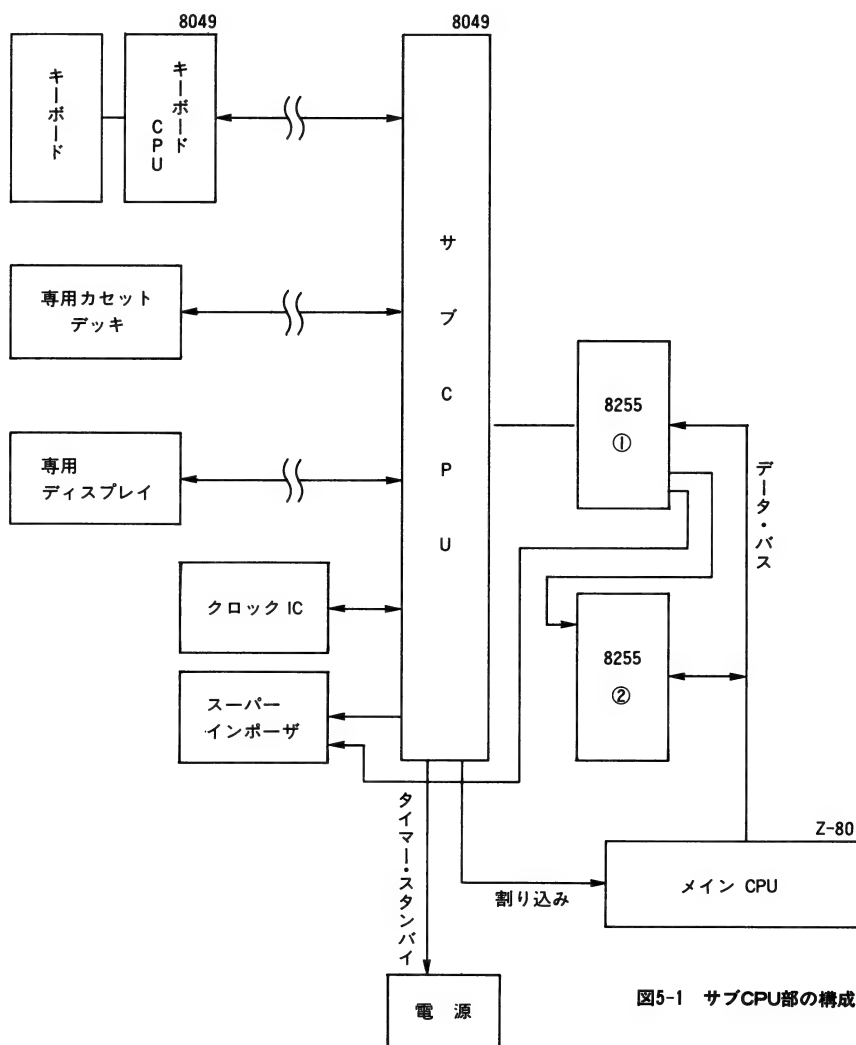


図5-1 サブCPU部の構成

ここで使われている CPU8049(80C49)は、8ビットのワンチップCPUで、2KバイトのROM、ワーク用のRAM、I/Oポートなどが内蔵されています。ROMには、サブCPUのコントロールプログラムが、ICの製造段階で書き込まれています。

ここでは、2つの8049を各々、キーボードCPU、サブCPUと呼ぶことにします(X1では、キーボードCPUは8048で、一部の機能に制限があります)。図5-1からもわかるように、これらのCPUは、以下のようなことを行っています。

(1) キーボードCPU

- ・キーボードをスキャンし、押されたキーのデータをサブCPUに送信

(2) サブCPU

- ・キーボードCPUからのデータ受信
- ・専用カセットデッキのコントロール
- ・タイマICの時刻データの読み書き
- ・メインCPUとのデータの受け渡し

サブCPUとキーボードCPUの電源は、メインCPU系統とは分離されています。フロントのスイッチがoffになっていても、後面パネルのスイッチが入っていれば、サブCPU系には電源が供給され、動作を続けます。このためフロントスイッチがoffであっても、キーボードを用いた各種のTVコントロールとTVタイマー機能は動作するのです。

サブCPUは、BASICを用いている限り、意識する必要はありません。しかし、マシン語を使って、I/Oを制御しようとするとしてもサブCPUとデータのやり取りを行わなければなりません。以降、サブCPUの使い方を説明していきます。

5-1-2 直接アクセスによるコマンドの送受信

サブCPUには、メインCPUと独立して動作しています。サブCPUに何かのコントロールを実行させたり、結果を受け取ったりするには、決められた手続きに従ってコマンドを送らなければなりません。サブとメインには、図5-1に示すように、2つの8255で接続されています。サブCPUとの通信に必要なI/Oポートは、以下に示す通りです。

・8255②ポートB(入力)

I/Oアドレス=1A01H

ビット6:IBF(Input Buffer Full)

Lの時8049にデータを送っても良い

ビット5: $\overline{\text{OBF}}$ (Output Buffer Full)

Lなら8049からのデータがある

・8255①ポートA(サブCPUの管理下にある)

I/Oアドレス=1900H

サブCPUとのデータの入出力

サブCPUと、直接データのやり取りをする場合は、データのぶつかり合いが起これないように、IBFと $\overline{\text{OBF}}$ の2本の制御線を用いて通信します。

(1) サブCPU にコマンドやデータを渡す場合

- 1) IBF を読み込む
- 2) IBF が1なら, 0になるまで待つ
- 3) データを1バイト 1900H に書き込む
- 4) まだ送るデータがあるなら1)へ

(2) サブCPU からデータを読み込む場合

- 1) $\overline{\text{OBF}}$ を読み込む
- 2) $\overline{\text{OBF}}$ が1なら, 0になるまで待つ
- 3) データを1バイト 1900H から読み込む
- 4) まだ読み込むべきデータがあるなら1)へ

サンプルプログラムをリスト5-1と5-2に示します。

リスト5-1 Z-80から80C49にデータを送る場合

```

GET49:  LD      BC, 01A01H
GET49_1: IN      A, (C)
        AND     040H
        JR      NZ, GET49_1
        LD      BC, 01900H
        LD      A, 0E3H
        OUT     (C), A
        RET
        ;
        END

```

80C49がデータを受け取れるまで待つ

ゲームキーデータ送信要求コマンド

リスト5-2 Z-80が80C49からデータを受け取る場合

```

RCV49:  LD      BC, 1A01H
R49_1:  IN      A, (C)
        AND     20H
        JR      NZ, R49_1
        LD      BC, 01900H
        IN      A, (C)
        RET
        ;
        END

```

80C49からデータが送られるまで待つ

データを受け取る

サブCPU との通信は、このように送る手順が決っており、バイト数はコマンドの種類によって変わってきます。従って、サブCPU の実行状態と、メインCPU の要求が何かのはずみで食い違くと、サブ-メイン間の通信が不可能となり、システムがハング・アップする可能性があります。プログラムの起動直後など、サブCPU の状態が不明の時は、サブCPU の初期化を行う必要があります。このためのサブルーチンを、リスト5-3に示します。

リスト5-3 80C49を初期化する

```

S49RES EQU      13E5H
INT49:  LD      BC, 1A01H
        IN      A, (C)
        AND     10H
        JR      NZ, C49_1
        LD      A, 1EH
        JR      C49_2
C49_1:  LD      A, 1EH
C49_2:  PUSH    AF

```

BIOS ROMがアクティブかどうか調べ、その状態をセーブする

L D	A, 1DH	ROMをアクティブに
O U T	(00H), A	
C A L L	S49RES	ROMの状態を元に戻す
P O P	A F	
O U T	(00H), A	
R E T		
:		
E N D		

5-2 キー入力

5-2-1 キー入力の概要

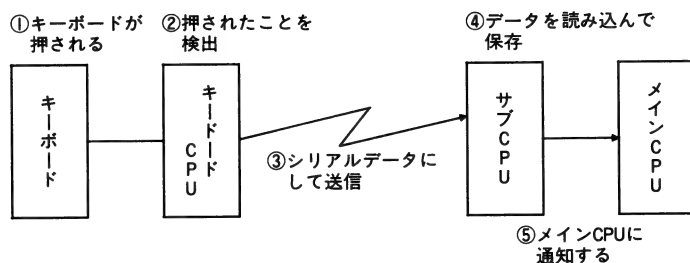


図5-2 キー・データの流れ

どのキーが押されたかというキーデータは、上の図に示すように伝達されます。キーボード用CPUは、すべてのキーを順次スキャンしており、新しく押されたキーや離されたキーを判定します。そして、キーの情報を直列のビット列にして、サブCPUに送ります。送る信号には、一般のキー入力に用いるモードA型の信号と、一度に複数のキーを読み取ることができるモードB型の信号があります。どちらの信号を使うかは、キーボード横のスイッチで選択します(X1には、モードBはありません)。サブCPUでは、送られてきた信号のパルスの幅から、モードを自動的に判別し、読み込みます。そして、その信号を基に現在のキーの状態を把握し、メインCPUからの要求があると、そのデータをメインCPUに渡します。

モードBで同時読み込みができるキーは、テンキーやスペースキーなどの24個のキーです。これを「ゲームキー」と呼びます。また、モードBでは、キーボードのカナ文字の配列が、JIS配列から五十音配列になります。

キーデータをメインCPUで受け取る方法には2通りあります。割り込み(インターラプト)を使用する方法と、使用しない方法です。割り込みを使うと、キーを押した瞬間に、データを受け取ることができるので、キー入力に対しすぐに応答しなければならない場合に適します。逆に、割り込みを使わない方法では、必要なときだけキーの情報を見に行けば良いので、不必要なキー入力を無視することができます。また、プログラムも若干簡単になります。

5-2-2 キーボードの信号

キーボードから送られて来る信号は、何種類もの幅を持ったパルス列による特殊なものです。この信号は、メインCPUから直接読むことはできないので、その概要だけ説明します。

(1) モード A 信号

キーボードからの信号は、図5-3のようになっています。ファンクションコード(1バイト目)は、データ(2バイト目)のデータの種類や各種シフトキーの内容を表しています。このファンクションコードの内容を、表5-1に示します。

モード A のキーデータ構成

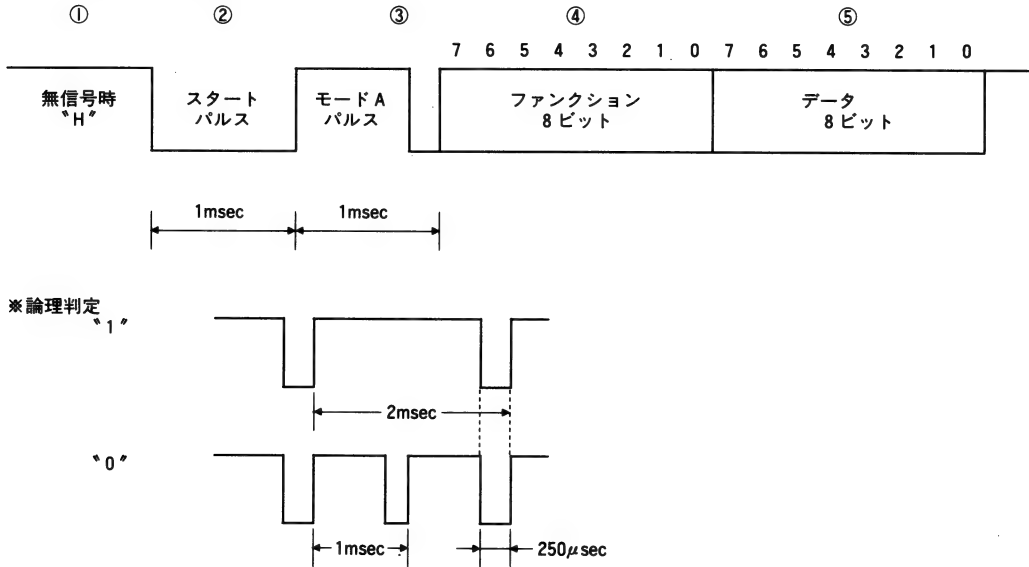


図5-3 モード A 信号

	(MSB) 7	6	5	4	3	2	1	(LSB) 0
	ファンクション	キーデータ有効/無効	リピート	GRAPH	CAPS	カナ	SHIFT	CTRL
0	・テンキー ・ファンクションキー ・TVキー ・カセットキー	・データ・コード(8ビット)が有効である。	・リピート・データである。	・GRAPHキーが押されている。	・CAPSキーが押されている。 (LOCKされている)	・カナキーが押されている。 (LOCKされている)	・SHIFTキーが押されている。	・CTRLキーが押されている。
1	・上記以外	・データ・コード(8ビット)が無効である。	・1回目のデータである。	・GRAPHキーがはなされている。	・CAPSキーがはなされている。	・カナキーがはなされている。	・SHIFTキーがはなされている。	・CTRLキーがはなされている。

表5-1 ファンクションコード

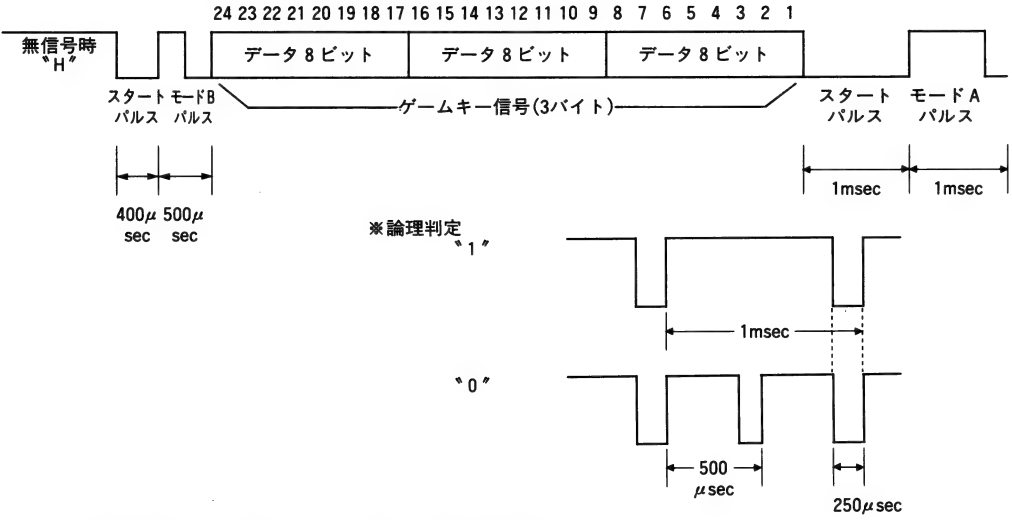
(2) モード B 信号

モードBでは、モードAで送られてくるのと同じキーデータの前に、3バイトのゲームキーデータが付くことがあります。この3バイト(24ビット)のゲームキーデータは、ゲームキーのいずれかが押されるか、離されたときに送られます。このデータによって、24種類のキーに限って同

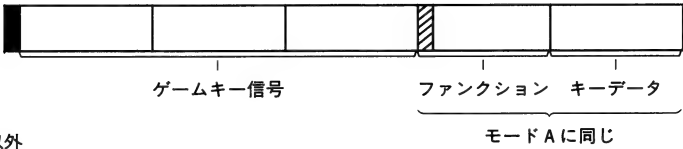
時に読み取ることができます。

図5-4にゲームキー信号の内容を示します。信号の各ビットは、24種類のそれぞれのキーに対応していて、そのキーが押されている時は、そのビットが1に、押されていないときは0になります。各ビットとキーの対応表を、表5-2に示します。

●ゲーム・キー信号の形式



●モード B 信号の形式 (1)ゲームキーが押された時/離された時



(2)それ以外

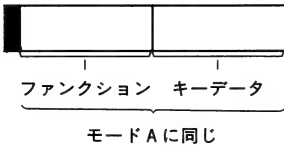


図5-4 モードB信号

ビット	1	2	3	4	5	6	7	8
キ ー	Q	W	E	A	D	Z	X	C

ビット	9	10	11	12	13	14	15	16
キ ー	7	4	1	8	2	9	6	3

テンキー

ビット	17	18	19	20	21	22	23	24
キ ー	ESC	/	-	+	*	HTAB	スペース	RET

テンキー

表5-2 ゲームキー信号の各ビットとキーの対応

5-2-3 割り込みを用いないキー入力

割り込みを用いるモードでも用いないモードでも、どちらのモードを使うかを、最初にサブCPUに宣言しておかなければなりません。このためのサブCPUのコマンドコードが、[E4H]です。

・サブCPU コマンド[E4H]

(キー入力割り込みのベクタアドレスセット)

メイン→サブ: E4, <アドレス>

サブ→メイン: なし

<アドレス> = 0 の時、割り込みを使わないモードになる。

割り込みを使わない場合は、0E4H に続いて 00H を送ります。そうするとサブCPUは、それ以降キー入力があっても、割り込みを起こしません。

実際にキーデータを読み込むには、コマンド[E6H]をサブCPUに送り、続いて2バイトサブCPUから読み込みます。この2バイトは、ファンクションコード1バイトと、押されたキーのASCII(アスキー)コード1バイトです。

・サブCPU コマンド[E6H]

(キーデータの読み込み)

メイン→サブ: E6

サブ→メイン: <ファンクションコード>, <キーコード>

<ファンクションコード>の内容は、表5-1に同じ

<キーコード>はASCIIコードに変換されている

リスト5-4 キー割り込みによらないキーデータ読み込みのサンプルプログラム

OT49SB EQU	1413H	
COMOUT EQU	1432H	
TAK49S EQU	143BH	
NKYIN: LD	BC, 1A01H	}
IN	A, (C)	
AND	10H	
JR	NZ, NKIN1	
LD	A, 01H	}
JR	NKIN2	
NKIN1: LD	A, 1EH	}
NKIN2: PUSH	AF	
LD	A, 1DH	}
OUT	(00H), A	
LD	A, 0E4H	}
CALL	COMOUT	
XOR	A	}
CALL	OT49SB	
LD	A, 0E6H	}
LD	DE, KEYBUF	
CALL	TAK49S	

BIOS ROMの状態を調べ、その状態をセーブする

ROMアクティブ

キー割り込みを禁止する

キー入力をバッファに受け取る

POP	AF] ROMの状態を元に戻す
OUT	(00H), A	
RET		
:		
KEYBUF: DS	2	
:		
END		

キーボードがモードBにセットされているときは、ゲームキーの状態を得ることができます。このときはコマンド[E3H]を使います。

・サブCPU コマンド[E3H]

(ゲームキーデータの読み込み))

メイン→サブ: E3

サブ→メイン: <ゲームキーデータ(3バイト)>

<ゲームキーデータ>の内容は、表5-2を参照

このコマンドは、キーボードがモードAでも動作しますが、返ってくるデータには意味がありません。必ずキーボードをモードBにして使用して下さい。

5-2-4 割り込みを用いたキー入力

X1シリーズでは、一般にZ-80CPUの割り込みモードをモード2にして使用します。モード2割り込みでは、割り込みが発生したときの処理ルーチンの先頭番地(エントリーアドレス)が書かれているアドレス(ベクタアドレス)を、CPUのIレジスタを上位8ビット、各I/Oデバイスの出力する値を下位8ビットとして指定します。そこでキー入力を割り込みで使うときは、割り込みベクタアドレスの下位8ビットを、あらかじめサブCPUにセットしておきます。このためのコマンドは、先ほど説明した[E4H]です。このコマンドに続いて、ベクタアドレスの下位8ビットを送ります。この値は0であってはなりません。0の場合、割り込みを使わない、という意味になってしまうからです。また、アドレスの最下位ビットは0でなければなりません。ベクタは、偶数番地から始めることになっているからです。

・サブCPU コマンド[E4H]

(キー入力割り込みのベクタアドレスセット)

メイン→サブ: E4, <アドレス>

サブ→メイン: なし

<アドレス>=割り込みベクタアドレスの下位8ビット

最下位ビットは必ず0でなければならない

<アドレス>=0の時、割り込みを使わないモードになる

割り込み処理ルーチンでは、5-2-3で示したのと同様、コマンド[E6H]を使ってキーデータ

を読み込みます。普通のプログラムでは、得られたキーデータをメモリ上のバッファにセットし、必要とあればキー入力フラグを立て、EI(割り込み許可)命令を実行してリターンします。Z-80では、割り込みがかかると、自動的に割り込み禁止になりますから、EIを忘れると二度と割り込みがかからなくなります。サンプルプログラムをプログラム5-5に示します。

リスト5-5 キー入力割り込みによるキーデータ読み込みのサンプルプログラム

```

IN49SB EQU      1408H
IKYIN:  PUSH     BC
        PUSH     DE
        PUSH     HL
        PUSH     AF
        LD       BC, 1A01H
        IN       A, (C)
        AND      10H
        JR       NZ, IKIN1
        LD       A, 1DH
        JR       IKIN2
IKIN1:  LD       A, 1EH
IKIN2:  PUSH     AF
        LD       A, 1DH
        OUT      (00H), A
        CALL     IN49SB
        LD       L, A
        CALL     IN49SB
        LD       H, A
        LD       (KEYBUF), HL
        POP      AF
        OUT      (00H), A
        POP      AF
        POP      HL
        POP      DE
        POP      BC
        EI
        RETI
;
KEYBUF: DS       2
;
END

```

BIOS ROMの状態をセーブ

ROMアクティブ

80C49よりキーデータを受け取り、バッファにセーブする

割り込み処理ルーチン内でゲームキーを読み込むときは、まず[E6H]のキー入力コマンドを実行し、キー入力データを受け取ってから[E3H]のゲームキー入力を行って下さい。

リスト5-6 キー割り込み中のゲームキー読み込み

```

IN49SB EQU      1408H
TAK49S EQU      143BH
GKYIN:  PUSH     BC
        PUSH     DE
        PUSH     HL
        PUSH     AF
        LD       BC, 1A01H
        IN       A, (C)
        AND      010H
        JR       NZ, GKIN1
        LD       A, 01DH
        JR       GKIN2
GKIN1:  LD       A, 1EH
GKIN2:  PUSH     AF

```

ROMの状態をセーブ

```

LD      A, 1DH
OUT     (00H), A
CALL    IN49SB
CALL    IN49SB
LD      A, 0E3H
LD      DE, GKBUF
CALL    TAK49S
POP     AF
OUT     (00H), A
POP     AF
POP     HL
POP     DE
POP     BC
EI
RETI
;
GKBUF:  DS      3
;
END

```

5-3 専用モニター TV のコントロール

5-3-1 モードの切り換え

X1 シリーズの専用モニターには次の4つのモードがあり、X1 本体からのリモコン信号によって切り換えることができます。

- 1) テレビ画面のみ
- 2) コンピュータ画面のみ
- 3) スーパーインポーズ1 (テレビとコンピュータ画面を重ねて表示、テレビのコントラストをダウンさせる)
- 4) スーパーインポーズ2 (テレビとコンピュータ画面を重ねて表示、テレビのコントラストはダウンさせない)

これらのモード切り換えも、サブCPU にコマンドを送ることによって行うことができます。このためのコマンドが[E7H]です。

・サブCPU コマンド[E7H]

(専用モニターのコントロール)

メイン→サブ: E7, <コード>

サブ→メイン: なし

モード切り換えのためのコードには、X1 シリーズ共通の1～5バイトのものと、X1turbo 以降追加された1バイトのものがあります。これらのコードを表5-3に示します。1～5バイトのコードでは、最初に0E7H, 05Hを送って、テレビ画面に切り換えてからモード切り換えをするので、一瞬テレビ画面が写ります。

(a) X1(1~5バイト)

画面モード \ 送信コード(バイト数)	1	2	3	4	5	6
T V 画面	E7	05				
コンピュータ画面	E7	05	E7	08		
スーパーインポーズ 1 (コントラストダウン)	E7	05	E7	0F	E7	0A
スーパーインポーズ 2 (コントラストノーマル)	E7	05	E7	0F		

コードはすべて16進数

(b) X1turbo以降(1バイト)

画面モード \ 送信コード(バイト数)	1	2
T V 画面	E7	1C
コンピュータ画面	E7	1D
スーパーインポーズ 1 (コントラストダウン)	E7	1E
スーパーインポーズ 2 (コントラストノーマル)	E7	1F

コードはすべて16進数

表5-3 モニターのモード切り換えコード

リスト5-7 モニター画面をコンピュータ画面にするためのアクセス手順

OT49SB EQU 1413H	
COMOUT EQU 1432H	
CMDSP: LD BC, 1A01H	ROMの状態をセーブ
IN A, (C)	
AND 10H	
JR NZ, CMDP1	
LD A, 1DH	
JR CMDP2	ROMをアクティブに
CMDP1: LD A, 1EH	
CMDP2: PUSH AF	80C49にコマンドとデータを送る。画面モードをコンピュータ画面に変更する
LD A, 1DH	
OUT (00H), A	
LD A, E7H	
CALL COMOUT	
LD A, 1DH	ROMを元の状態に戻す
CALL OT49SB	
POP AF	
OUT (00H), A	
RET	
; END	

5-3-2 コントロール

専用モニターでは、モード切り換えの他、チャンネル、音量などのコントロールがコンピュータ側から可能です。この制御にも、コマンド[E7H]を使います。このコマンドの後に、表5-4に示すコードを送ります。

内 容	送信コード(バイト数)	
	1	2
ボ リ ュ ー ム ア ッ プ	E7	01
ボ リ ュ ー ム ダ ウ ン	E7	02
ボリュームノーマル(42/64階調)	E7	03
音 声 ミ ュ ー ト	E7	06
チャ ン ネ ル ア ッ プ	E7	0B
チャ ン ネ ル ダ ウ ン	E7	0C
パ ワ ー オ フ	E7	0D
パワーオン/オフ(トグル動作)	E7	0E
チャ ン ネ ル 1	E7	10
チャ ン ネ ル 12		1B
パ ワ ー オ ン	E7	80

コードはすべて16進数

表5-4 モニターのコントロールコード

リスト5-8 モニター画面をスーパーインポーズ1モードにするためにアクセス手順

OT49SB EQU 1413H

COMOUT EQU 1432H

I1DSP: LD BC, 1A01H

IN A, (C)

AND 010H

JR NZ, I1DP2

LD A, 1DH

JR I1DP2

I1DP1: LD A, 1EH

I1DP2: PUSH AF

LD A, 1DH

OUT (00H), A

LD A, E7H

CALL COMOUT

LD A, 1EH

CALL OT49SB

POP AF

OUT (00H), A

RET

;

END

ROMの状態をセーブ

ROMをアクティブに

画面モードをスーパーインポーズ1に変更

ROMを元の状態に戻す

5-4 専用カセットデッキのコントロール

5-4-1 カセットメカのコントロール

X1シリーズの専用カセット(内蔵のものを含む)は、サブCPUによってコントロールされています。メカのコントロールの他、カセットの有無、テープエンド、消去防止ツメなどのチェックもしています。カセットのコントロールをメインCPUから行うには、コマンド[E9H]を使います。なお、サブCPUが行うのは、メカニズム関係のみで、実際の信号の読み書きはメインCPUから直接I/Oポートを通じて行います。8255②の項(5-6-3)を参照して下さい。

・サブCPU コマンド[E9H]

(カセットメカのコントロール)

メイン→サブ：E9, <コントロールコード>

サブ→メイン：なし

コントロールコードは1バイトです。その内容を表5-5に示します。ここでAPSSとは頭出しのための状態で、ヘッドをテープにつけたまま早送りや巻き戻しを行います。このときのAPSS読み取り信号は、8255①(サブCPU側)に接続されています。

動 作	コントロールコード
EJECT	00
STOP	01
PLAY	02
FF	03
REW	04
APSS-FF	05
APSS-REW	06
REC	0A

表5-5 カセット・コントロール・コード

5-4-2 ステータスの読み出し

専用カセットを使用しているときは、サブCPUから現在のカセットの状態を知ることができます。このためのコマンドとして、メカの状態を知る[EAH]とカセットの状態を知る[EBH]があります。

・サブCPU コマンド[EAH]

(カセットメカニズムの状態検出)

メイン→サブ:EA

サブ→メイン:＜状態コード＞

＜状態コード＞の見方は、表5-5を参照

・サブCPU コマンド[EBH]

(カセットの状態検出)

メイン→サブ:EB

サブ→メイン:＜カセット状態コード＞

＜カセット状態コード＞の見方は、図5-5を参照

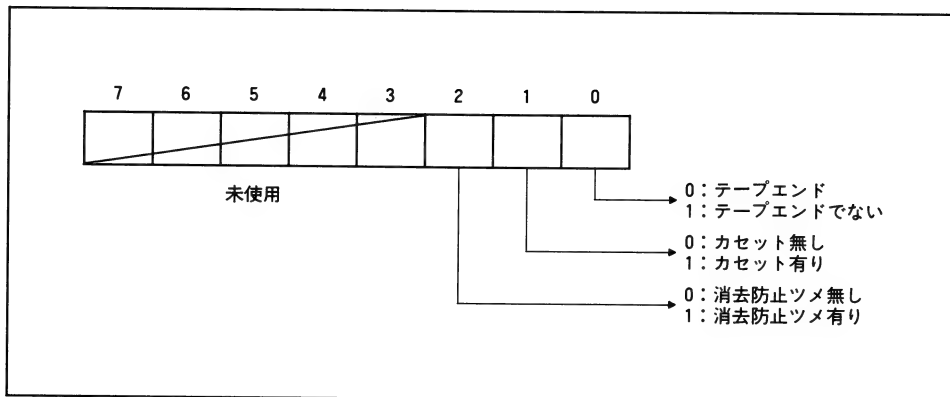


図5-5 カセット状態コード

カセットメカが動作中(PLAYやRECの状態など)にBREAKキー等が押されると、サブCPUは次のような動作を行います。

(1) BREAK キーが押された時

- ・カセットをSTOPする
- ・メインCPUのBREAKフラグ(8255②PB-0)をLにする
- ・キー入力割り込みをメインCPUにかける(割り込みが許可されている時)
- ・メインからのキー入力に対しては、キーコード03Hを返す

(2) カセットコントロールキーが押された時

- ・押されたキーに対応するカセットの動作を行う
- ・メインCPUのBREAKフラグ(8255②PB-0)をLにする

リスト5-10 カセットメカのコントロール

```

OT49BS EQU      1413H
COMOUT EQU      1432H
TCCTR:  LD      BC, 1A01H
        IN      A, (C)
        AND     10H
        JR      NZ, TCCT1
        LD      BC, 1D00H
        JR      TCCT2
TCCT1:  LD      BC, 1E00H
TCCT2:  LD      A, 1DH
        OUT     (00H), A
        LD      A, 0E9H
        CALL    COMOUT
        LD      A, (01H)
        CALL    OT49SB
        OUT     (C), A .....ROMを元に戻す
        RET
        ;
END

```

ROMの状態をセーブ

ROMをアクティブ

カセットストップコード送信

リスト5-11 カセットメカの状態の読み出し

```

IN49SB EQU      1408H
COMOUT EQU      1432H
RCCTR:  LD      BC, 1A01H
        IN      A, (C)
        AND     10H
        JR      NZ, RCCT1
        LD      BC, 1D00H
        JR      RCCT2
RCCT1:  LD      BC, 1E00H
RCCT2:  LD      A, 1DH
        OUT     (00H), A
        LD      A, 0EAH
        CALL    COMOUT
        CALL    IN49SB
        LD      (RCCBF), A
        OUT     (C), A .....ROMを元に戻す
        RET
        ;
RCCBF:  DS      1
        ;
END

```

ROMの状態をセーブ

ROMをアクティブに

カセット状態読み出し

5-4-3 カセットテープのフォーマット

カセットに記録されているデータの読み書きは、メインCPUがI/Oポートを通して直接行います。カセット関係のI/Oポートは、以下の通りです。

・8255 ②ポート B(入力)

I/Oアドレス=1A01H

ビット1: READ DATA

カセットからの読み込み信号

・8255 ②ポート C(出力)

I/Oアドレス=1A02H

ビット0: WRITE DATA

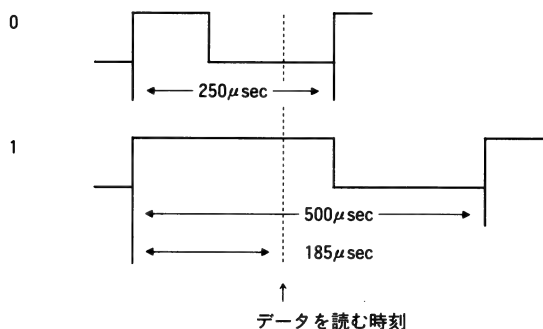
カセットへの書き込み信号

カセット書き込み用のポートC出力は、他のコントロール出力と共通ですから、カセット出力に必要なビット0以外は変更しないように注意しなければなりません。このためには8255のビットセット・リセット機能を使用します。

さて、X1シリーズの標準フォーマットでは、信号にシャープPWM方式という変調方式が使用されています。PWM(Pulse Width Modulation:パルス幅変調)では、パルスの幅を変化させて情報を記録します。パルスの1サイクルが $250\mu\text{s}$ なら0、 $500\mu\text{s}$ なら1となっています。読み込むときは、波形がHになってから $185\mu\text{s}$ 後の状態を調べ、このときHなら1、Lなら0と判断します。

読み書きは1バイトを単位として行われます。パルスの形式と、1バイトの構成を図5-6に示します。

● 1 ビットの信号



● 1 バイトの構成

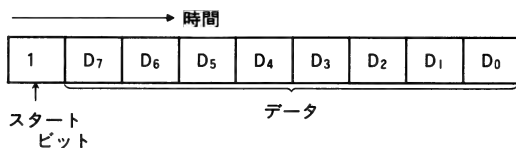
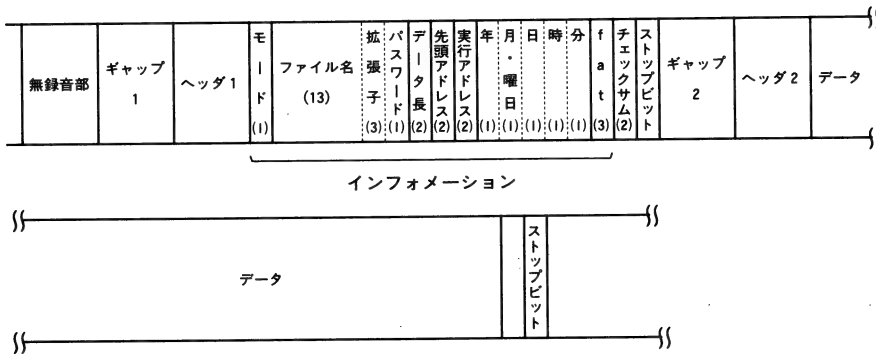


図5-6 シャープPWM方式

X1シリーズのテープフォーマット(論理フォーマット)は、インフォメーションブロックとデータブロックの2つに大きく分けられます。各ブロックの最後にはデータの総和を取った2バイトのチェックサムがつきます。全体では、まず8秒間の無録音部分、インフォメーションブロック、そしてデータブロックと続きます。また、データブロックについては、データが連続してベタで記録される「連続セーブ」と256バイトずつのブロック毎に区切って記録する「ブロッキングセーブ」があります。

BASICでは、プログラムのセーブに連続セーブ、データファイルやASCIIセーブにはブロッキングセーブを使っています。フォーマットの詳細を図5-7に示します。

(1) 連続セーブの場合



(2) ブロッキングセーブの場合

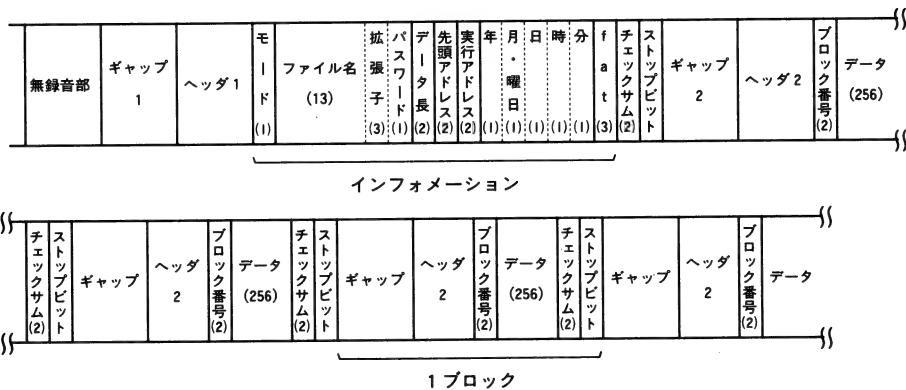


図5-7 テープの論理フォーマット

5-5 タイマーのコントロール

5-5-1 時刻の設定と読み出し

サブCPUには、タイマーIC(μ PD1990)が接続されており、日付や時刻の読み出し、書き込みをメインCPUから行うことができます。タイマーICは、ニッカド(充電式)電池でバックアップされており、本体の電源が切れても時を刻み続けます。ただし、「年」の情報だけはサブCPUがカウントしているため、本体背面のメインスイッチを切ると「年」情報は無くなってしまいます。

日付、時刻の読み出し、書き込みのサブCPUコマンドは次の通りです。

・サブCPUコマンド[ECH]

(日付の設定)

メイン→サブ: EC, <日付データ(3バイト)>

サブ→メイン: なし

・サブCPU コマンド[EDH]

(日付データの読み出し)

メイン→サブ：ED

サブ→メイン：＜日付データ(3バイト)＞

・サブCPU コマンド[EEH]

(時刻の設定)

メイン→サブ：EE, ＜時刻データ(3バイト)＞

サブ→メイン：なし

・サブCPU コマンド[EFH]

(時刻データの読み出し)

メイン→サブ：EF

サブ→メイン：＜時刻データ(3バイト)＞

日付データ、時刻データはどちらも3バイトのブロックになっています。これらのデータの構成を次頁に示します。

リスト5-12 カレンダー時計から日付と日時を読み出してメモリーに書き込む

```

TAK49S EQU      143BH
RCCLD:  LD       BC, 1A01H
        AND      10H
        JR       NZ, RCCL1
        LD       BC, 1D00H
        JR       RCCL2
RCCL1:  LD       BC, 1E00H
RCCL2:  PUSH     BC
        LD       A, 1DH
        OUT      (00H), A
        LD       DE, CLBUF
        LD       A, 0EDH ..... 日付読み出しコマンド
RCCL3:  PUSH     AF
        CALL    TAK49S
        INC     DE
        INC     DE
        INC     DE
        POP     AF
        ADD     A, 02H ..... =EF 時刻読み出しコマンド
        JR     NC, RCCL3
        POP     BC
        OUT     (C), A
        RET
        ;
CLBUF:  DS       6
        ;
        END

```

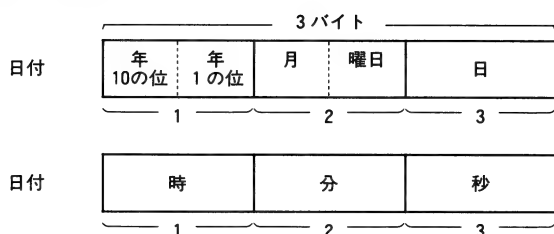
ROMの状態をセーブ

ROMをアクティブに

データをバッファにセーブ

ROMを元の状態に戻す

●データの形式



項 目	デ ー タ 説 明								
年	ビット	7	6	5	4	3	2	1	0
	<div><div></div><div></div><div></div><div></div></div>				<div><div></div><div></div><div></div><div></div></div>				
	10の位				1の位				
	(注1) 00年～99年の値をそのまま指定して下さい。								
月、 曜日	ビット	7	6	5	4	3	2	1	0
	<div><div></div><div></div><div></div><div></div></div>				*	<div><div></div><div></div><div></div></div>			*印は無効 ビット
	月				曜日				
	(注2) 月：(1)H(1月)～(C)H(12月)までの値を指定して下さい。(0)Hを指定すると無効(XX)となります。 曜日：(0)H(日)～(6)H(土)までの値を指定して下さい。								
日	ビット	7	6	5	4	3	2	1	0
	<div><div></div><div></div><div></div><div></div></div>				<div><div></div><div></div><div></div><div></div></div>				
	10の位				1の位				
	(注3) 10の位：(0)～(3)Hまでの値を指定して下さい。 1の位：(1)～(9)Hまでの値を指定して下さい。								
時	ビット	7	6	5	4	3	2	1	0
	<div><div></div><div></div><div></div><div></div></div>				<div><div></div><div></div><div></div><div></div></div>				
	10の位				1の位				
	10の位：(0)～(2)Hまでの値を指定して下さい。 1の位：(0)～(9)Hまでの値を指定して下さい。								
分	ビット	7	6	5	4	3	2	1	0
	*	<div><div></div><div></div><div></div></div>			<div><div></div><div></div><div></div></div>			*印は無効 ビット	
	10の位			1の位					
	(注4) 10の位：(0)～(5)Hまでの値を指定して下さい。 1の位：(1)～(9)Hまでの値を指定して下さい。								
秒	ビット	7	6	5	4	3	2	1	0
	*	<div><div></div><div></div><div></div></div>			<div><div></div><div></div><div></div></div>			*印は無効 ビット	
	10の位			1の位					

図5-8 日付、時刻データの構成

5-5-2 テレビタイマーの設定と読み出し

X1シリーズには、最大7回まで設定できるテレビタイマーコントロール機能があります。これらはサブCPUが行っているため、本体のフロント電源が off でも動作します。

テレビタイマーに関するサブCPU コマンドコードは、[D1H]～[D7H](設定)と[D9H]～[DFH](読み出し)の14個で、コードによって7つのタイマの選択を行います。

・サブCPU コマンド[D1H]～[D7H]

(テレビタイマーの設定)

メイン→サブ：[コマンド], <タイマーデータ(6バイト)>

サブ→メイン：なし

・サブCPU コマンド[D9H]～[DFH]

(テレビタイマーの設定状態の読み出し)

メイン→サブ：[コマンド]

サブ→メイン：<タイマーデータ(6バイト)>

コマンドの一覧表を表5-6に、タイマーデータの構成を図5-9と表5-7に示します。

TIMER番号	設定コード	読み出しコード
1	D1	D9
2	D2	DA
3	D3	DB
4	D4	DC
5	D5	DD
6	D6	DE
7	D7	DF

コードはすべて16進数

図5-6 タイマー番号と送信コードの対応

6バイト					
コントロール対象 及びインターバル	コントロール内容	分	時	月 曜日	日
1	2	3	4	5	6

図5-9 タイマーデータの構成

項 目	デ ー タ 説 明																																																								
コントロール対象及びインターバル	ビット 7 6 5 4 3 2 1 0 <table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> — タイマー有無 — — インターバル(1～59分) — <table border="1"><tr><th colspan="2">ビット内容</th><th rowspan="2">機 能</th></tr><tr><th>7</th><th>6</th></tr><tr><td>0</td><td>0</td><td>タ イ マ ー 無 効</td></tr><tr><td>1</td><td>0</td><td rowspan="2">未 使 用</td></tr><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>タ イ マ ー 有 効</td></tr></table> <p>※インターバルタイマーとは、あるタイマーが動作してから指定された時間間隔(1～59分)が経過したら、再び同じ動作を行うタイマーです。</p>									ビット内容		機 能	7	6	0	0	タ イ マ ー 無 効	1	0	未 使 用	1	1	0	1	タ イ マ ー 有 効																																
	ビット内容		機 能																																																						
	7	6																																																							
0	0	タ イ マ ー 無 効																																																							
1	0	未 使 用																																																							
1	1																																																								
0	1	タ イ マ ー 有 効																																																							
コントロール内容	7 6 5 4 3 2 1 0 <table border="1"><tr><td></td><td>0</td><td>0</td><td></td><td></td><td></td><td></td><td></td></tr></table> ↓ — システムビット — — TVコントロール内容コード — <table border="1"><tr><td>0</td><td>TV ONのコードは出力しない。</td></tr><tr><td>1</td><td>TVパワーONしたのち、0～4ビットのコードをTVへ送ります。</td></tr></table> <p>全コントロール内容とその設定データコードを下表に示します。 希望のコードを設定して下さい。(00)Hの場合、タイマー動作をしません。</p> <table border="1"><thead><tr><th>コントロール内容</th><th>データコード(Hex)</th></tr></thead><tbody><tr><td>ボリュームアップ</td><td>01</td></tr><tr><td>ボリュームダウン</td><td>02</td></tr><tr><td>ボリュームノーマル</td><td>03</td></tr><tr><td>音声ミュート</td><td>06</td></tr><tr><td>チャンネルアップ</td><td>0B</td></tr><tr><td>チャンネルダウン</td><td>0C</td></tr><tr><td>パワーオフ</td><td>0D</td></tr><tr><td>パワーオン／オフ (トグル動作)</td><td>0E</td></tr><tr><td>チャンネル 1</td><td>10</td></tr><tr><td> {</td><td>{</td></tr><tr><td>チャンネル 12</td><td>1B</td></tr><tr><td>パワーオン</td><td>80</td></tr><tr><td>パワーオン→ボリュームアップ</td><td>81</td></tr><tr><td>パワーオン→ボリュームダウン</td><td>82</td></tr><tr><td>パワーオン→ボリュームノーマル</td><td>83</td></tr><tr><td>パワーオン→音声ミュート</td><td>86</td></tr><tr><td>パワーオン→チャンネルアップ</td><td>8B</td></tr><tr><td>パワーオン→チャンネルダウン</td><td>8D</td></tr><tr><td>パワーオン→チャンネル 1</td><td>90</td></tr><tr><td> {</td><td>{</td></tr><tr><td>パワーオン→チャンネル 12</td><td>9B</td></tr></tbody></table>		0	0						0	TV ONのコードは出力しない。	1	TVパワーONしたのち、0～4ビットのコードをTVへ送ります。	コントロール内容	データコード(Hex)	ボリュームアップ	01	ボリュームダウン	02	ボリュームノーマル	03	音声ミュート	06	チャンネルアップ	0B	チャンネルダウン	0C	パワーオフ	0D	パワーオン／オフ (トグル動作)	0E	チャンネル 1	10	{	{	チャンネル 12	1B	パワーオン	80	パワーオン→ボリュームアップ	81	パワーオン→ボリュームダウン	82	パワーオン→ボリュームノーマル	83	パワーオン→音声ミュート	86	パワーオン→チャンネルアップ	8B	パワーオン→チャンネルダウン	8D	パワーオン→チャンネル 1	90	{	{	パワーオン→チャンネル 12	9B
		0	0																																																						
	0	TV ONのコードは出力しない。																																																							
	1	TVパワーONしたのち、0～4ビットのコードをTVへ送ります。																																																							
	コントロール内容	データコード(Hex)																																																							
	ボリュームアップ	01																																																							
	ボリュームダウン	02																																																							
	ボリュームノーマル	03																																																							
	音声ミュート	06																																																							
	チャンネルアップ	0B																																																							
チャンネルダウン	0C																																																								
パワーオフ	0D																																																								
パワーオン／オフ (トグル動作)	0E																																																								
チャンネル 1	10																																																								
{	{																																																								
チャンネル 12	1B																																																								
パワーオン	80																																																								
パワーオン→ボリュームアップ	81																																																								
パワーオン→ボリュームダウン	82																																																								
パワーオン→ボリュームノーマル	83																																																								
パワーオン→音声ミュート	86																																																								
パワーオン→チャンネルアップ	8B																																																								
パワーオン→チャンネルダウン	8D																																																								
パワーオン→チャンネル 1	90																																																								
{	{																																																								
パワーオン→チャンネル 12	9B																																																								

分	ビット 7 6 5 4 3 2 1 0 * 10の位 1の位 *印は無効ビット (注1) 設定値として(00)H～(7F)Hまで取り得ますが、(00)H～(59)H(0分～59分)以外の値を指定した場合、そのタイマー動作について保障できません。この項を無効にはできません。																		
時	ビット 7 6 5 4 3 2 1 0 10の位 1の位 (注2) 設定値として(00)H～(FF)Hまで取り得ますが、(FF)Hを書き込んだ場合のみ、自動的に時間の項が無効(XX)となります。また25H(25時)以上の値を指定した場合、そのタイマー動作について保障できません。																		
月, 曜日	ビット 7 6 5 4 3 2 1 0 月: 1～12月 曜日 *印は無効ビット (注3) 月: 13[(D)H]未満のBCD値を指定する。(0)Hのときは無効(XX)となります。曜日: 全ビットが"0"になる(7)H又は(F)Hを指定すると、自動的に曜日の項が無効(XXX)になります。 <table><tr><th>設定値 (H)</th><th>曜 日</th></tr><tr><td>0 or 8</td><td>SUN</td></tr><tr><td>1 or 9</td><td>MON</td></tr><tr><td>2 or A</td><td>TUE</td></tr><tr><td>3 or B</td><td>WED</td></tr><tr><td>4 or C</td><td>THU</td></tr><tr><td>5 or D</td><td>FRI</td></tr><tr><td>6 or E</td><td>SAT</td></tr><tr><td>7 or F</td><td>XXX</td></tr></table>	設定値 (H)	曜 日	0 or 8	SUN	1 or 9	MON	2 or A	TUE	3 or B	WED	4 or C	THU	5 or D	FRI	6 or E	SAT	7 or F	XXX
設定値 (H)	曜 日																		
0 or 8	SUN																		
1 or 9	MON																		
2 or A	TUE																		
3 or B	WED																		
4 or C	THU																		
5 or D	FRI																		
6 or E	SAT																		
7 or F	XXX																		
日	ビット 7 6 5 4 3 2 1 0 10の位 1の位 (注4) BCDの値を指定します。(32)H以上の値を書き込めますが、そのときのタイマー動作は保障できません。(00)Hの場合、日の項は無効(XX)となります。																		

表5-7 タイマーコントロールバイト(6バイト)の各ビットの内容

リスト5-13 テレビタイマーの設定例

TAK49S EQU 143BH	
STTMR: LD BC, 1A01H] BIOS ROMの状態をセーブ
AND 10H	
JR NZ, STTM1	
LD BC, 1D00H	
JR STTM2	
STTM1: LD BC, 1E00H] ROMアクティブ
STTM2: PUSH BC	
LD A, 1DH	
OUT (00H), A	

```

LD      DE, TMBUF
LD      A, 0D1H
CALL    TAK49S
POP     BC
OUT     (C), A
RET
;
TMBUF:  DB      1 .....タイマー番号
        DB      40H, 96H
        DB      17H, 00H, 7DH, 26H
        ;
END

```

5-5-3 タイマー用 IC

X1 シリーズに使われている μ PD1990 は、カレンダー機能、クロック機能を持った IC です。サブ CPU とは、40ビットのシリアルデータの形でデータの入出力を行います。この IC は、充電式電池によってバックアップされており、本体の電源を切っても動作します。電池の充電は、サブ CPU の電源系統を使って行われており、従ってフロントの電源スイッチが off であっても、背面のメインスイッチが入っていれば、電池の充電が行われます。

この IC は、データの読み書きをしていない時、DATA OUT 端子から 1Hz の信号を出力しています。この信号は、画面のブリンク用に使われています。

5-6 PPI(8255)

X1 シリーズでは、メイン CPU とサブ CPU に 1 つずつ、PPI と呼ばれる I/O デバイスが接続されており、メイン \longleftrightarrow サブ CPU 間の通信や、各種の I/O 制御に使われています。サブ CPU に接続された方の PPI を 8255①、メイン CPU に接続された方を 8255②と呼びます。

5-6-1 PPI の概要

8255PPI は、インテル社が開発し、世界で広く使われているパラレル入出力用の LSI です。この LSI には 24 本の入出力端子(ポート)があり、これらは 8 本ずつ 3 組に分けてポート A、ポート B、ポート C と呼ばれます。各ポートは、入力にするか出力にするか、データ転送にハンドシェイク法を使うかどうか、などが CPU からのプログラムによって決められるようになっています。また、データが入力された時に割り込みをかけたり、ポート C の出力を 1 ビットずつセット、リセットする機能もあります。図 5-10 に、8255②の機能設定などに使われるコントロールレジスタの内容を示します。なお、8255①はサブ CPU の管理下あるのでメイン CPU 側から操作することはできません。

8255の初期設定は、X1 シリーズの場合、リセット時に IPL によって行われています。初期設定は、ハードウェアと密接な関係があり、設定を勝手に変更したりしてはいけません。

さて、8255には大きく分けて 3 つのモードがあります。

・モード 0

単純な入力か出力しかしないモードです。8 ビットの入出力ポート 2 個(ポート A、ポート B)と、4 ビットの入出力ポート 2 個(ポート C の上位 4 ビット、下位 4 ビット)をそれぞれ、入力にするか出力にするか決めることができます。

・モード1

8ビットのデータを、3本のコントロールラインを使ったハンドシェイクと呼ぶ方法で転送します。コントロールラインにはポートCの一部が使われます。

・モード2

モード2はポートAだけが指定できます。このモードを指定するとポートAは、8ビットの入出力兼用線となります。ポートCの内5本がコントロールラインとなり、ハンドシェイクによるデータ転送を制御します。サブCPUとのコマンド送受信の項で、IBF(Input Buffer Full)、OBF(Output Buffer Full)というフラグが出てきましたが、実はこの5本のコントロールラインの一部なのです。

X1シリーズでは、サブCPUの管理下にある8255①のポートAがモード2に設定され、メインCPUとの通信に使われている他は、すべてモード0の単純入出力になっています。入力、出力の設定の様子を表5-9に示します。先に述べたように、これらの設定をプログラムで勝手に変更してはいけません。従ってコントロールワードのうちポートCのビットセット、リセット機能だけが使用できます。

グループ	ポート端子	アクティブ	コントロール内容	信号名
A	PA7	—	データ入出力	1D7
	PA6	—		1D6
	PA5	—		1D5
	PA4	—		1D4
	PA3	—		1D3
	PA2	—		1D2
	PA1	—		1D1
	PA0	—		1D0
	PC7	L	Z-80Aに対してデータ受取り	OBF
	PC6	L	Z-80AがポートAからデータを受取り信号	8049RD
	PC5	H	Z-80Aに対してデータ転送禁止信号	1BF
	PC4	L	Z-80AからのデータをポートAに入力/ラッチ指示信号	8255WE
B	PC3	—	未使用	
	PC2	H	PLAY時READ LED点灯します(L:WRITE LED)	
	PC1	L	Z-80AへのBREAK信号	BREAK
	PC0	L	カセットのEJECTソレノイドコントロール	EJECT SOL
	PB7	—	OBF信号	
	PB6	—	8049RD信号	
	PB5	—	APSS(無記録部検出)	APSS DATA(注)
	PB4	L	EJECT SW. センス	EJECT SW
	PB3	—	未使用	
	PB2	H	カセットテープの書き込み禁止用の爪がある状態	REC PROTECT
	PB1	H	カセットがセットされている状態	PACKAGE
	PB0	L	テープエンド検出	TAPE END

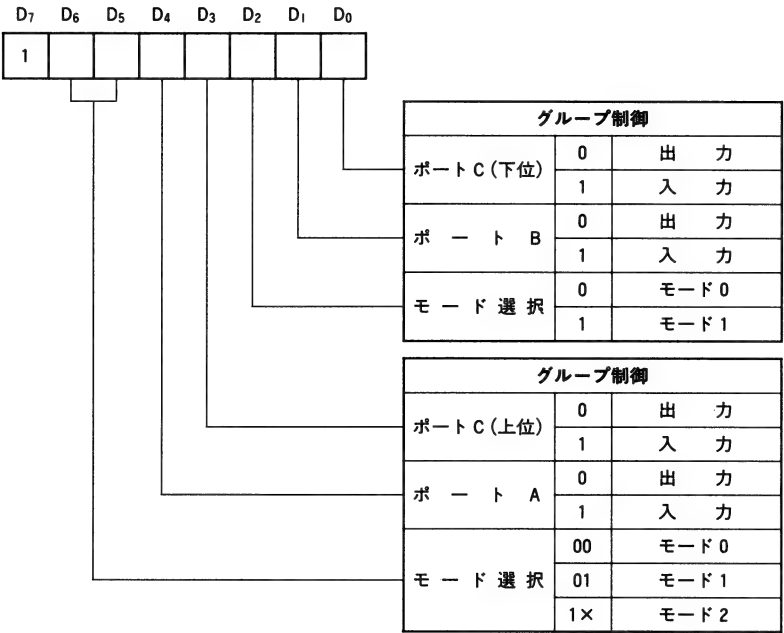
(注) READ DATA信号を、積分回路を通すことにより得た信号。

表5-8 X1 turbo model 10 における8255①のポートの割り当て

	CPU	ポート	モード	設 定
8255①	サ ブ	A B C	2 0 —	入出力 入 力 入出力
8255②	メイン	A B C	0 0 —	出 力 入 力 出 力

表5-9 X1シリーズにおける8255の設定

●モード設定



●ビット・セット/リセット

コントロールワード

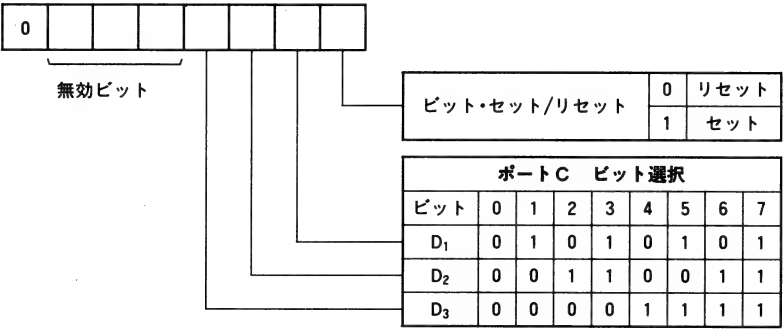


図5-10 8255のコマンドレジスタ

第 6 章

割り込み

6-1 割り込み処理の概要

割り込みは、I/O からの処理要求が発生したら、その時実行していたプログラムを中断（インターラプト）し、必要な処理に移る機能です。割り込みを上手に活用することによって、非常に効率のよいプログラムを作成することができます。

割り込み処理は BASIC でも一部サポートされていることはご存じでしょう。

```
1000 'サンプル
1010 ON KEY GOSUB 2000
1020 KEY 1 ON
1030 'メインルーチン
1040 ....
1050 ....
1060 ....
.
.
1500 GOTO 1030
2000 'KEY INT ROUTINE
2010 XX=POS(0):YY=CSRLIN
2020 LOCATE0,0:PRINT"STATUS:";STA
2030 LOCATE XX,YY
2040 RETURN
```

上のプログラムではメインルーチンの処理中に[F1]キーが押されると画面の上隅に変数 STA の内容を表示します。この場合、キーボードが「割り込み(をかける)デバイス」、[F1]キーの押下が「割り込み要因」、行番号2000からが「割り込み処理ルーチン」となります。プログラムの最初で[F1]割り込み処理ルーチンの行番号「割り込みベクタ」を宣言し(ON KEY GOSUB)、続いて割り込みを許可します(KEY 1 ON)。Z80 の割り込み処理についてこのあと説明しますが、概念としてはこの BASIC の処理にたいへん似ています。

X1 シリーズには、キーボードやディスクを始めとして数多くの I/O がありますが、そのほとんどは割り込みが使用できるように設計されています。しかも Z80 の割り込み処理の方法のうち、最も強力なモード 2 割り込みが使用できます。そこで、最初に Z80 の割り込み処理について説明しましょう。

6-2-1 Z80 の割り込み処理

Z80CPU は、INT と名付けられた端子があり、この端子を L レベルにすることで割り込み処理が始まります。もう少し正確に言うと、1つの命令を実行し終った時点で INT 端子が L であり、かつ割り込み許可の状態であれば CPU は「ある方法で」割り込み処理ルーチンを call するのです。この時の call の方法にはモード 0～モード 2 の 3 種類があります。どのモードを使うかは予めプログラムで設定しておきます (IM0～IM2 命令)。

①モード 0

このモードは、Z80 の前身とも言える、インテル社の 8080A CPU とほぼ同じ動作をします。

CPU は、通常はプログラムカウンタ PC の指す番地のメモリの内容を読み込み、解釈し、実行する、ということを繰り返しています。これに対しモード 0 で割り込みが起これば CPU は、メモリへの読み出し命令を出さずに命令を読み込み、それを実行します。そこでハードウェアを工夫し割り込みが発生した瞬間、適当な命令 (通常は RST0～7 の call 命令) を CPU に読み込ませるようにします。call 命令の飛び先に割り込み処理ルーチンを書いておきます。

②モード 1

このモードの動作は簡単です。割り込みが発生すると、CPU は自動的に RST7 を実行します。すなわち現在の PC の値をスタックに PUSH して 0038H 番地にジャンプします。

モード 1 では割り込み処理ルーチンの入口が 1 つしかないため、複数の割り込みデバイスがある場合、どこから割り込みがかかったかの判定はソフトウェアで行います。例えばキーボード、プリンター、タイマーの 3 つの割り込みデバイスがあるときは図 6-1 のように処理します。この場合、複数のデバイスが同時に割り込みを起こした時、どれを最初に実行するか (割り込みの優先順位) は、割り込み要因を調べる順番で決まります。先に調べられるデバイスの方が優先的に処理されます。

③モード 2

割り込み優先順位の決定から、各装置ごとの処理ルーチンの call までを自動的に行うようにしたのがモード 2 です。3 つのモードの中では最も強力で、ソフトウェアでさまざまな判断をする必要がないので割り込みに対する応答が早くなります。X1 シリーズでは、もっぱらこのモード 2 が使用されています。

モード 2 で割り込みが発生すると、CPU は「ベクタ」と呼ばれる 1 バイトの情報を読み込みます。そして、このベクタを下位、CPU 内部の I レジスタの内容を上位とするアドレスから 16 ビットのアドレスを読み込み、そこを call します。この様子を 図 6-2 に示します。ここでは、I レジスタの値は 0ABH、割り込みデバイスである PIO の出した (あらかじめ PIO にセットしておいた) ベクタの値が 0CDH だったので CPU は 0ABCDH の内容を読み込みます。それが 1234H であったので、CPU は call 1234H を実行します。

結局、I レジスタが上位、00H を下位とする番地から 256 バイトが割り込み処理ルーチンのテーブルとなり、受け付けられる割り込みの種類は最大 128 種類、ということになります。

割り込み処理ルーチンは、各レジスタの PUSH から始めます。割り込み処理によってレジスタが破壊されると、もとの処理に戻った時に暴走する原因となるからです。処理が終了したら、PUSH したレジスタをもとに戻し、EI (割り込み許可)、RETI (割り込みからのリターン) の順番に実行します。EI を実行するのは、割り込みがかかると自動的に DI (割り込み禁止) 状態になるため、EI を実行しないと 2 回目の割り込みがかかなくなるからです。また、RETI を実行すると割り込みをかけたデバイスは、処理終了とみなして割り込み要求を解除するようになっています。

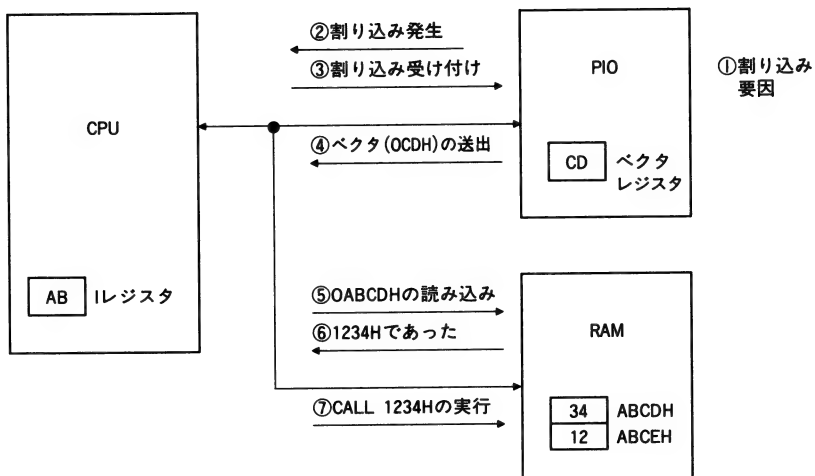
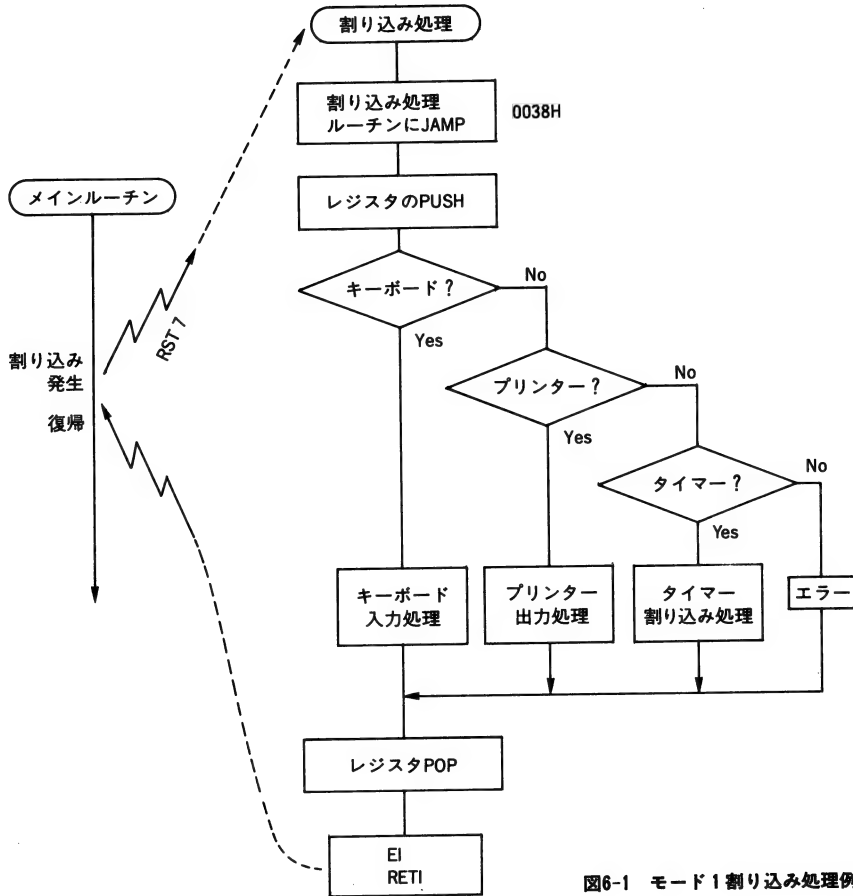


図6-2 モード2 割り込み処理

6-1-2 メイン CPU の割り込みデバイス

先に述べたように、X1 シリーズでは、Z80CPU の割り込みモードのうち、最も強力なモード 2 割り込みが使用できるように設計されています。割り込みが使用できるデバイスには、以下のものがあります。

1. キー入力(サブ CPU)
2. Z80-SIO(RS-232C, マウス)
3. Z80-CTC
4. Z80-DMA
5. 拡張 I/O スロット(外部割り込み)

割り込みの優先順位はハードウェアで決められており、以下のようになっています。

I/O スロット 1 > I/O スロット 2 > SIO > DMA > CTC > キー入力

左側のデバイスほど優先的に処理されます。なお、X1 では、内部割り込みデバイスはキー入力だけであり、拡張スロットは 1～4 の 4 つになります。優先順位は、次のようになります。

I/O スロット 1 > I/O スロット 2 > I/O スロット 3 > I/O スロット 4 > キー入力

割り込み時のベクタの値は、予め各デバイスにセットしておかなければなりません。ベクタの一覧を表 6-1-1 に示します。個々のデバイスの使い方については、次節以降で説明します。

	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SIO	*	*	*	*	↓	↓	↓	0
(チャンネル B)	送信バッファ室	0	0	0				
外部/ステータスの変化	0	0	1					
受信キャラクタ有効	0	1	0					
特別受信条件	0	1	1					
(チャンネル A)	送信バッファ室	1	0	0				
外部/ステータスの変化	1	0	1					
受信キャラクタ有効	1	1	0					
特別受信条件	1	1	1					
DMA	*	*	*	*	*	↓	↓	0
• RDY割り込み						0	0	
• 一致割り込み						0	1	
• エンド・オブ・ブロック割り込み					1	0		
• エンド・オブ・ブロック 一致割り込み					1	1		
CTC	*	*	*	*	*	↓	↓	0
• チャンネル	0	0	0					
• //	1	0	1					
• //	2	1	0					
• //	3	1	1					
KEY	*	*	*	*	*	*	*	0

• 但し、00Hは使用できない

*：任意の値に設定できる

↓：割り込み条件によって自動的に決まる

0：最下位ビットはゼロでなければならない

表6-1 各種割り込みデバイスのベクタアドレスの下位バイトの構成

6-2 シリアルI/O

X1シリーズにおけるシリアルI/Oは、RS-232C インターフェイスとマウスインターフェイスです。これらシリアルI/Oには、Z80-SIO という非常に高機能なインターフェイス LSI が使用されています。

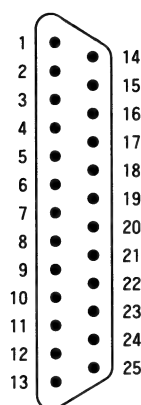
6-2-1 シリアルI/Oの概要

パソコン用としてシリアルI/Oが活用されはじめたのは最近のことです。いわゆる「パソコン通信」の発展とともに、これからはますますその重要性を増していくことと思われます。最初にシリアルI/Oで出てくるいくつかの用語について説明しましょう。

① RS-232C インターフェイス

本来はEIA(アメリカ電子工業会)が、DCE(回線接続装置)と端末間のインターフェイス条件について定めた規格です。RS-232Cでは、コネクタのピン配置と信号の電氣的な仕様が定められています。但し、信号の形式については何も定められていません。232Cを使用するときに非常に多くの設定をしなければならないのは、このためでもあります。

RS-232Cのピン配置を表6-2に示します。この中でパソコンレベルで実際に使われているのは、GND(グラウンド)と入出力線、及びRTS/CTSのハンドシェイク線くらいです。RTS/CTSすら使わず、3本の線だけで接続することも多いようです。232Cの信号は、 $\pm 10V$ 以上の電圧を持っており、そのままではパソコン内部のTTLレベルのLSIに接続することができません。このため専用のインターフェイス用ICが使われます。



ピン番号	内 容
1	Protective Ground
2	Transmitted Data
3	Received Data
4	Request to Send
5	Clear to Send
6	Data Set Ready
7	Signal Ground (Common Return)
8	Received Line Signal Detector
9	(Reserved)
10	(Reserved)
11	Unassigned
12	Sec. Rec'd. Line Sig. Detector
13	Sec. Clear to Send
14	Secondary Transmitted Data
15	Transmission Signal Element Timing (DCE Source)
16	Secondary Received Data
17	Receiver Signal Element Timing (DEC Source)
18	Unassigned
19	Secondary Request to Send
20	Data Terminal Ready
21	Signal Quality Detector
22	Ring Indicator
23	Data Signal Rate Selector (DTE/DCE Source)
24	Transmit Signal Element Timing (DTE Source)
25	Unassigned

表6-2 RS-232Cのピン配置

②調歩同期式

パソコンでRS-232C インターフェイスの使われ方としては、パソコン同士を接続してのデータ転送、ネットワークにつなぐためのモデムとの接続、ミニコン等の端末として、といった使い方が多いと思います。これらの場合、300～9600ボー程度の調歩同期式通信が使われます。

この方法では、1本の信号線で直列に1ビットづつ信号を送っていきます。何も信号が無いとき、ラインはマーク(H)であり、そこに図6-3のように1文字分の信号をのせます。各信号の意味は、次の通りです。

- ・スタートビット：信号の始まりを示す。1ビット分の長さの「0」。
- ・文字ビット：5～8ビットのデータ。
- ・パリティ：1ビットの誤りを発見することができる。
偶数パリティ、奇数パリティ、パリティ無しのいずれか。
- ・ストップビット：1, 1.5, 2のいずれかの長さの「1」。

ここで、「1ビットの長さ」を示すのにボーレート(Baud Rate)という言葉が使われます。ボーレートが9600ボーであるとは、信号をぎっしり詰めて送ると、1秒間に9600ビット分ある、という意味です。実際にはスタートビットなどの無駄な信号がありますから、データの転送速度はこれより小さくなります。

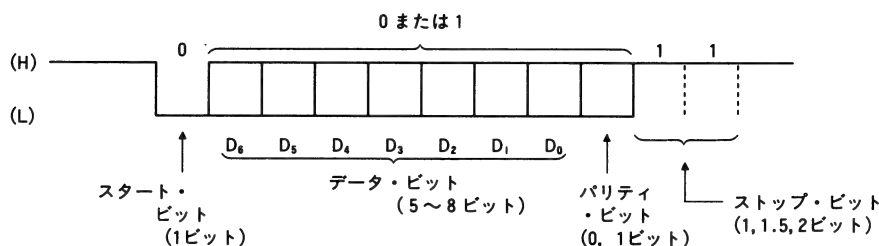


図6-3 調歩同期通信のデータフォーマット

ある、パソコンネットワークの規格は次のようになっていました。

2400ボー、 8ビットデータ、 パリティ無し、
1ストップビット、XON/XOFF 制御有り、

今までの説明で、最初の4項目の意味はわかるでしょう。最後のXON/XOFFとは、データの転送停止、再開をコントロールする方式の1つです。データを受け取る方の処理が間に合わなくなりそうになったら、相手方にXOFFキャラクタ(^S)を送り、転送を一時停止してもらい、処理が終了したらXONキャラクタ(^Q)を送って再開します。

③SDLC, HDLC

データを1文字ずつ送る従来の方法が持っていたいくつかの欠点を解決し、高速で、使いやすく、信頼性の高い方式として制定されたのがHDLC(High-level Data Link Control)方式です。この規格の基となったIBMの開発した方式がSDLC方式で、X1シリーズに使われている

Z80-SIOはこのSDLCをサポートしています。

SDLCは、任意の長さのデータと制御記号から成る「フレーム」を単位として行われます。細かい解説は省略しますが、この方式は以下のような特徴を持っています。

- ・バイナリデータを送ることが容易。
- ・任意の長さのデータをまとめて送るので転送速度が速い。
- ・CRCを使った誤り訂正制御が行われるので、信頼性が高い。
- ・訂正不能の誤りが発生した時の再送シーケンスについても厳密に定められており、それらの制御も連続的に行われるので伝送効率が高い。

6-6-2 SIO まわりの構成

X1シリーズでは、SIOのチャンネルAはRS-232C、チャンネルBはマウスの制御に使われています。SIOまわりの構成を図6-4に示します。

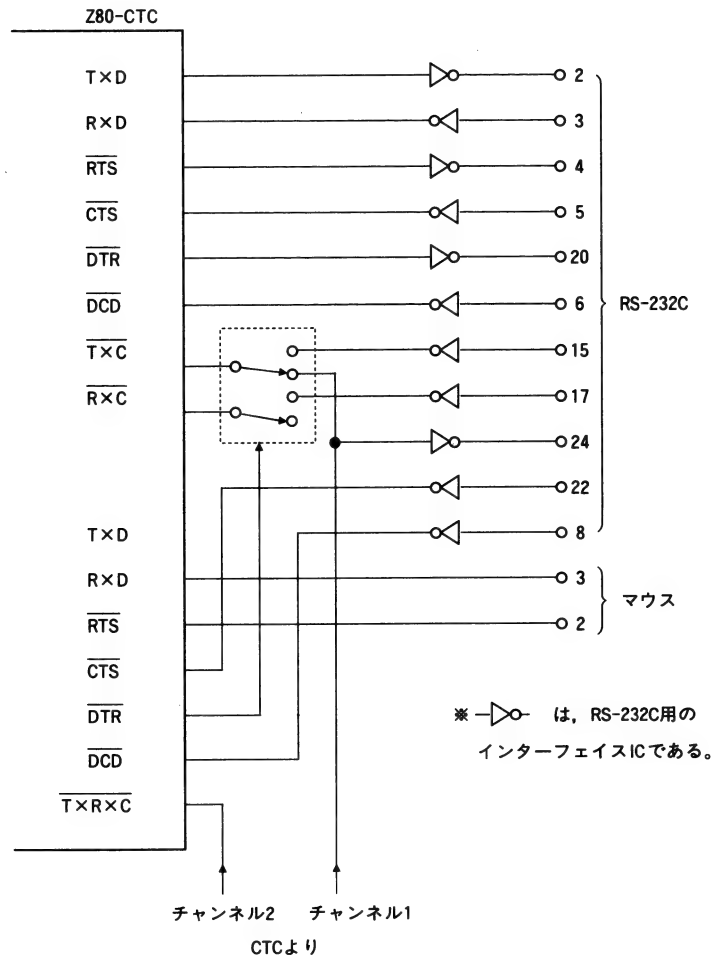


図6-4 SIOまわりの構成

RS-232C には多くの制御線がありますが、X1 シリーズでサポートされているのはその一部です。サポートされる制御線と、SIO の端子との関係は、以下の通りです。

SIO端子	チャンネル	IN/OUT	RS-232C端子 (ピン番号)
TxD	A	OUT	Transmitted Data (2)
RxD	//	IN	Received Data (3)
RTS	//	OUT	Request to Send (4)
CTS	//	IN	Clear to Send (5)
DTR	//	OUT	Data Terminal Ready (20)
DCD	//	IN	Data Set Ready (6)
TxC	//	IN	Transmission Signal Element (15)/CTC ch-1 output. (*)
RxC	//	IN	Received Signal Element (17)/CTC ch-1 output. (*)
DCD	B	IN	Received Line Signal (8)
CTS	//	IN	Ring Indicator (22)
GND			Frame GND (1)
GND			Signal GND (7)

(*) チャンネル B の DTR が「H」の時 CTC に接続される

表6-3 SIO と RS-232C の関係

パソコン同士を接続して通信するときは、

RD (3) ↔ TD (2)
 RTS (4) ↔ CTS (5)
 DTR (20) ↔ DCD (6)

の各線をひっくり返しにして接続すれば、通信できます。但し、この場合上記 6 つの信号線以外の入力データは意味がありません。

SIO のポートアドレスを表 6-4 に示します。このようにチャンネル A とチャンネル B のそれぞれについてデータポートと制御語(コントロールポート)があります。SIO には制御のための書き込みレジスタが各チャンネルに 8 個ずつあります。レジスタに値を書き込むときには、まずレジスタ番号をコントロールポート(書き込みレジスタ 0)にセットし、続いて目的のレジスタの値を書き込みます。例えばレジスタ 2 に 10H を書き込むときはコントロールポートに 02H, 10H と書き込みます。但し、書き込みレジスタ 0 については 1 回で書き込みます。

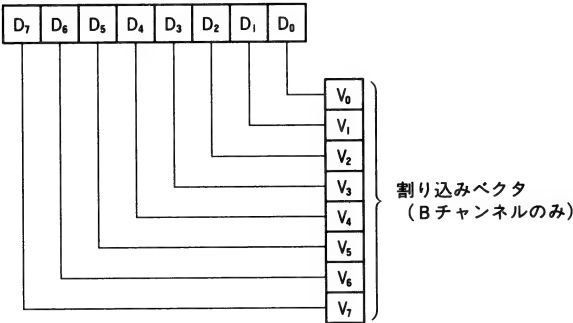
また、内部状態を知るための読み出しレジスタの内容を読むときも、最初にレジスタ番号を書き込み、続いて読み出しを行います。読み出しレジスタ 0 だけは直接読み出せます。

書き込み、読み出しの各レジスタの内容を次頁から示します(表 6-5)。SIO は SDLC をサポートしているため、レジスタの内容はたいへん複雑になっています。

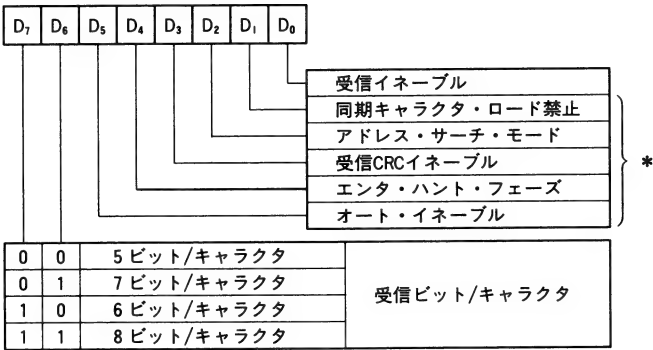
アドレス	ポート内容
1F90H	チャンネル A データポート
1F91H	チャンネル A 制御語
1F92H	チャンネル B データポート
1F93H	チャンネル B 制御語

表6-4 SIO の I/O ポート

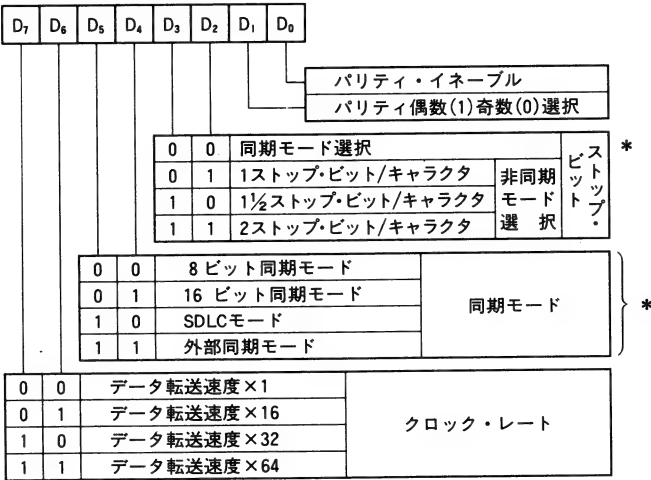
書き込みレジスタ 2 : WR2



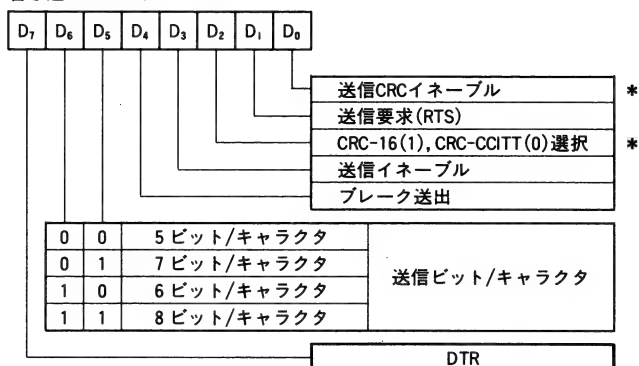
書き込みレジスタ 3 : WR3



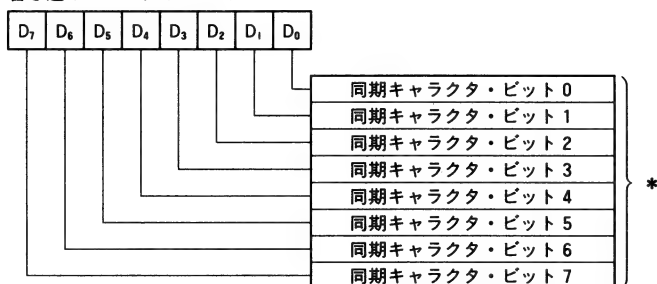
書き込みレジスタ 4 : WR4



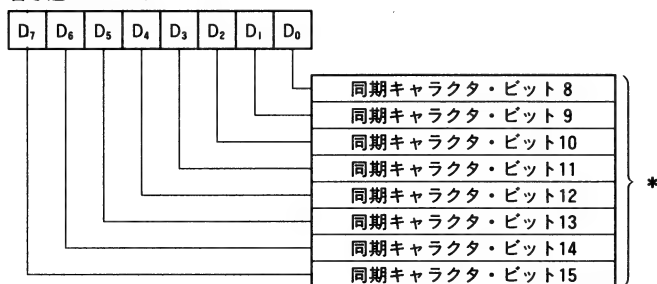
書き込みレジスタ 5 : WR 5



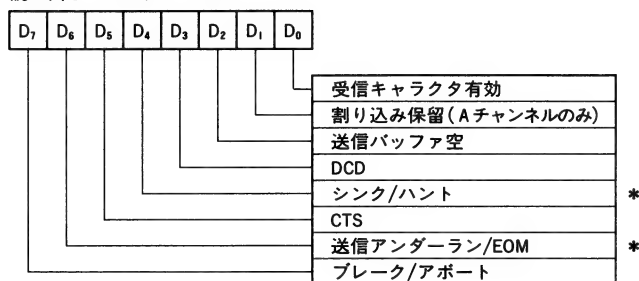
書き込みレジスタ 6 : WR 6



書き込みレジスタ 7 : WR 7



読み出しレジスタ 0 : RRO



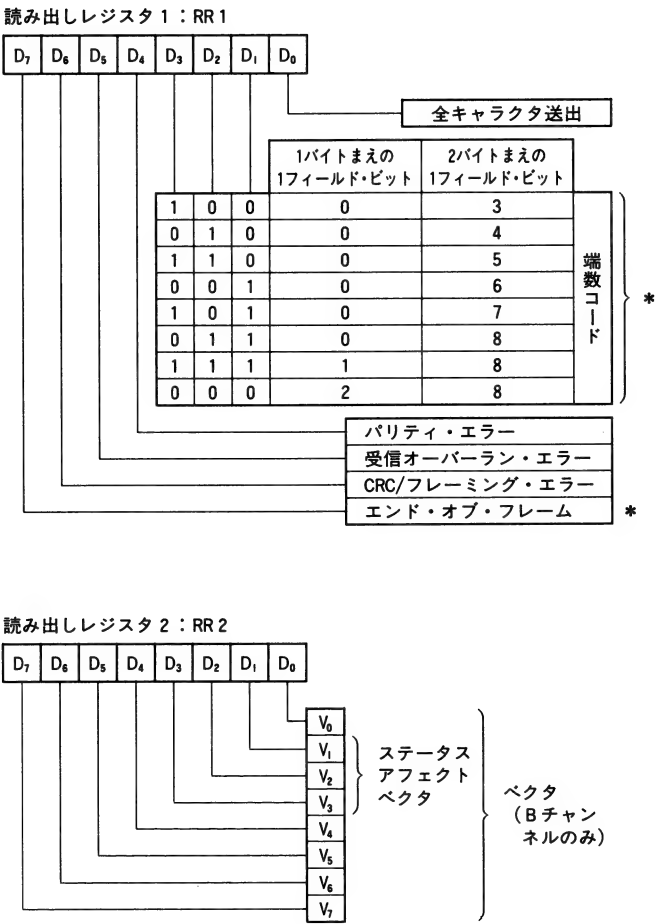


表6-5 Z80-SIOのレジスタ構成

6-2-3 チャンネル A(RS-232C)

チャンネルAは、RS-232Cの入出力として使用されています。

通信のためのクロックは、内部同期か外部同期かを選択することができます。この選択にはチャンネルBのDTR端子を使用します。外部同期にするとときはここを0とし、内部同期の時は1にします。

内部同期の時のボーレートは、CTCのチャンネル1の分周比設定とSIOの設定の両方で決まります。ボーレートとこれらの設定の関係を、下に示します。

CTC分周比	SIO設定	
	1/16	1/64
13	9600	2400
26	4800	1200
52	2400	600
104	1200	300
208	600	150

* CTC分周比=52, SIO設定=1/64ならば600ボーとなる。 表6-6 ボーレートの設定

BIOS ROM には、RS-232C 関係の処理ルーチンが用意されていますから、これらを利用すると入出力が簡単にできます。しかし、ハードウェアによるハンドシェイクなどを行うときは、その処理はユーザープログラム側で行わなければなりません。もっとも、先に述べた通り、ハードウェアハンドシェイクが使われることはパソコンレベルではあまりありません。

6-2-4 チャンネル B(マウス)

SIO のチャンネル B はマウスの制御に割り当てられています。使われている端子は 2 つだけです。マウスへのコントロール出力(CTRL)としての RTS 出力と、マウスからのデータを受け取る RxD 入力です。

マウスは RTS 端子が H から L になると、3 バイトのデータを送ってきます。送られてくるデータは次のようなフォーマットになっています。

4800 ボー、8 ビットデータ、パリティなし、1 ストップビット

従って、SIO もこの通り設定します。ボーレートは CTC のチャンネル B を使って RS-232C と同様に設定します。4800 ボーと決まっていますから、CTC の分周比は 26、SIO の設定は 1/16 とします。

マウスからのデータは 3 バイトで、1 バイト目がステータス、2 バイト目と 3 バイト目が各々 X、Y 方向の前回からの移動量(−128~127)です。ステータスの内容は次の通りです。

データ	内 容
D0	スイッチ 1 の状態を示します。 0 =OFF, 1=ON
D1	スイッチ 2 の状態を示します。 0 =OFF, 1=ON
D2	_____
D3	_____
D4	オーバーフロービット, X が 128 以上の時 1
D5	アンダーフロービット, X が −129 以下の時 1
D6	オーバーフロービット, Y が 128 以上の時 1
D7	アンダーフロービット, Y が −129 以下の時 1

表6-7

リスト6-1 マウスからのデータの読み込み

SIOBAD EQU	1F94H	
RDMSE: LD	BC, SIOBAD	
LD	HL, RDMDT	
LD	A, 05H	SIOのチャンネルBのRTS端子を' L 'にする
OUT	(C), A	
INC	HL	
LD	A, 0E0H	
OUT	(C), A	
LD	D, 03H	
RDMS1: IN	A, (0E0H)	
RRC	A	
JR	NC, RDMS1	
LD	A, 01H	SIOからデータを受信する
OUT	(C), A	
IN	A, (C)	
LD	E, A	
DEC	C	
INC	HL	
IN	A, (C)	
INC	C	
LD	(HL), A	

```

          LD      A, E
          AND     70H
          JR      NZ, RDMS3
          DEC     D
          JR      NZ, RDMS1
          XOR     A
RDMS2:    RET
RDMS3:    LD      A, 0FFH
          JR      RDMS2
          ;
RDMDT:    DS      3
          ;
          END

```

] マウスの移動量が-129～+128をこえた

6-3 DMA

6-3-1 DMA の概要

DMA(Direct Memory Access)とは、CPU を介さずにハードウェアで直接データを転送することです。メモリやI/Oの間でデータの転送を行うとき、DMAを使用すると非常に高速に転送を行うことができます。X1turbo以降にはZ80-DMAと呼ばれるLSIが実装されており、フロッピーディスクの読み書きやV-RAMのスクロールなどに使われています。

DMAに指令を与えるには、SIOなどと同じようにOUT命令を使用します。実行命令が与えられるとDMAは、CPUを停止させ、CPUの代わりに各種の制御信号を出力しながらデータの転送を行います。従ってその動作は一般のI/Oとはかなり異なります。

Z80-DMAの特徴は次の通りです。

1. メモリ↔メモリ、I/O↔I/O、I/O↔メモリのいずれの転送も可能。
2. データ転送のほか、データサーチやサーチしながらの転送が可能。
3. 転送データの長さは、2バイト～64Kバイト。
4. 転送が終了するまでCPUを止める、CPUと並列に転送するなど、4つのモードを持つ。
5. 転送アドレスは、固定、インクリメント、デクリメントのいずれか。
6. 転送速度は最高500Kバイト/秒以上、サーチ速度は最高1000Kバイト/秒以上(但し条件によって大きく異なる)。

CPUのプログラムの実行と、DMAのデータ転送のタイミングはDMAの転送モードで決まります。転送モードには次の4種類があります。

1. コンティニュアスモード：転送がすべて終了するまでCPUは停止する。
2. バーストモード：RDY信号(データ転送可能信号)が入力されている間CPUは停止する。転送可能でないときはCPUが動作する。
3. バイトモード：1バイト転送すると、CPUが動作する。データが連続的にくる場合は、DMAとCPUは交互に動作することになる。
4. トランスペアレントモード：CPUのメモリリフレッシュサイクルを使って、CPUの動作とDMAが同時に実行される。

X1turboでは、フロッピーディスクの読み書きにバイトモードが、V-RAMのスクロールやその他のメモリ転送にバーストモードが、それぞれ使用されています。なお、トランスペアレントモードは、X1シリーズでは使用できません。

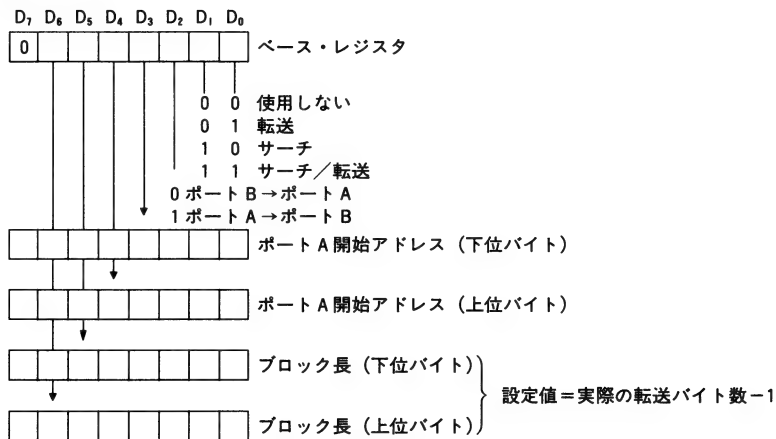
6-3-2 DMA の使い方

DMA には、書き込みレジスタが21個、読み出しレジスタが7個あります。

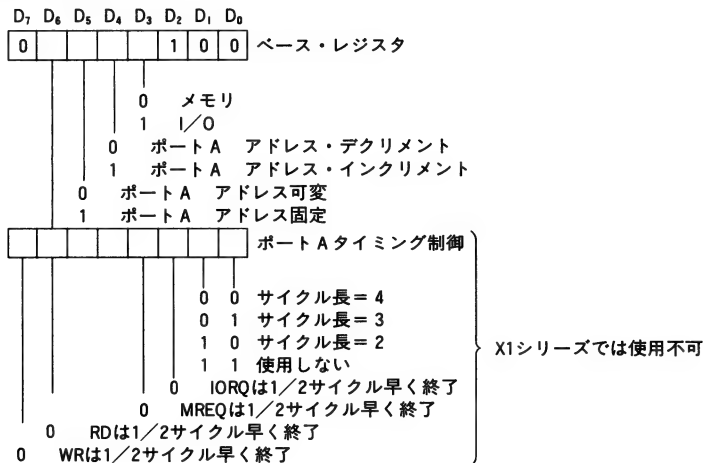
●書き込みレジスタ

21個の書き込みレジスタは、さらに7個のベースレジスタと21個の関連レジスタに分けられます。ベースレジスタはリセット時には内容は不定ですので、必ず7個とも初期化しなければなりません。関連レジスタは、必要なもののみ初期化します。これら書き込みレジスタの内容一覧を、図6-3-1に示します。

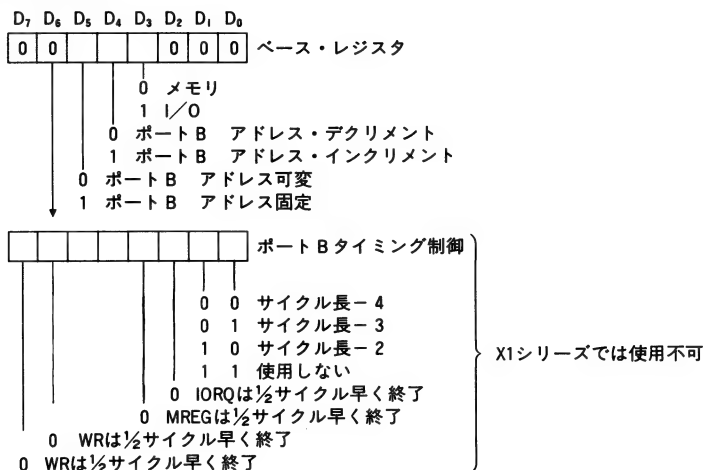
書き込みレジスタ 0



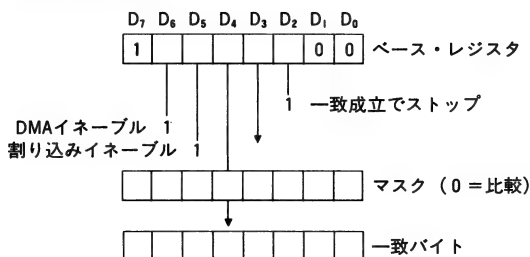
書き込みレジスタ 1



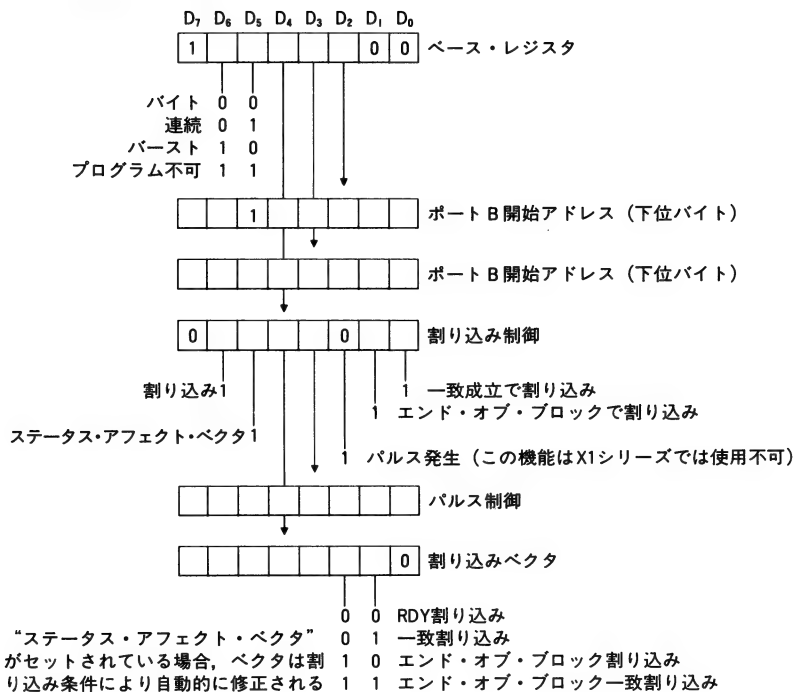
書き込みレジスタ 2



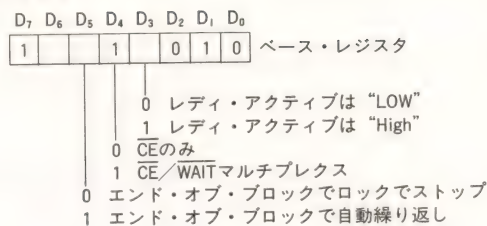
書き込みレジスタ 3



書き込みレジスタ 4



書き込みレジスタ 5



書き込みレジスタ 6

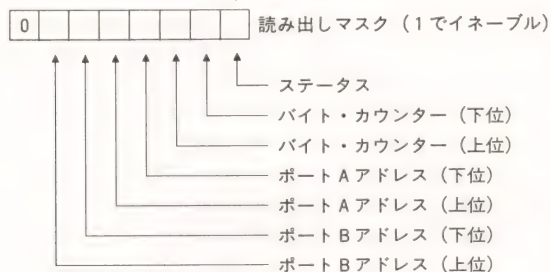
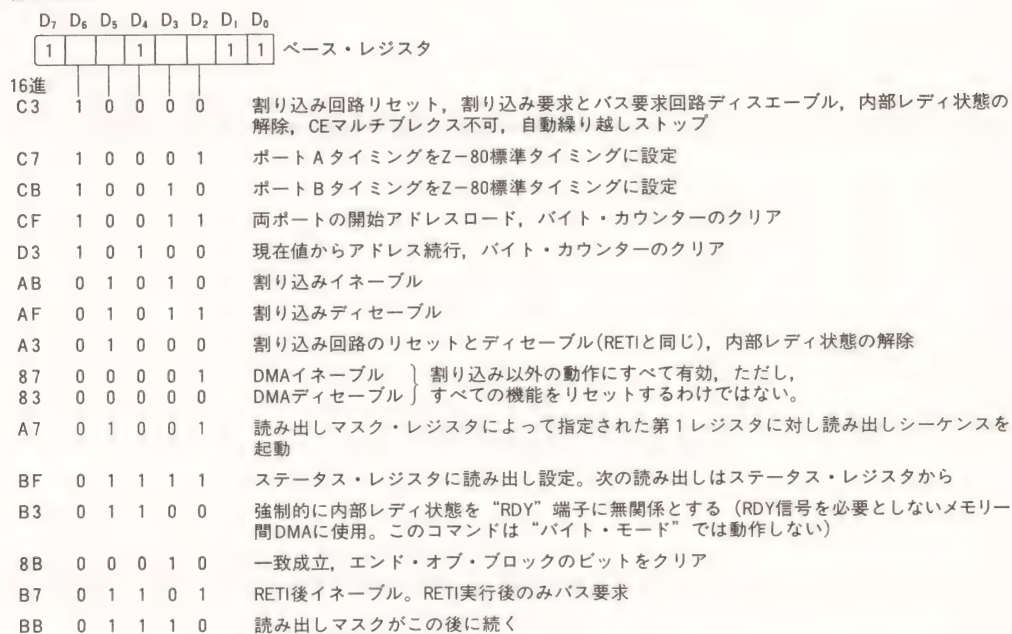


図6-5 Z80-DMA書き込みレジスタ

ベースレジスタに値を書き込むには、I/O アドレス 1080H に対して OUT 命令を実行します。I/O アドレスが1つしかないのに、どうやって7つのレジスタを区別するのかわかる人いるでしょう。もう一度図 6-5 を見て下さい。各レジスタの中に、0か1かが決められているビットや、「使用しない」となっているデータパターンがあることに気が付くと思います。そして、さらに良く観察すると、データのビットパターンから、どのレジスタへの書き込みかが決定でき

ることがわかります。実はDMAは、書き込まれたデータのパターンからどのレジスタへの書き込みかを判断しているのです。

関連レジスタへデータを書き込むときは、ベースレジスタの所定のビットを1にして書き込み、それに続いて関連レジスタを書き込みます。複数の関連レジスタを指定した時は、図の順番に従って、続いて書き込みます。

DMAは、最終的に87H(DMAイネーブル)を書き込むと動作を開始し、転送が終了すると動作終了となります。また、バイトモードやバーストモードでは、CPUがDMAに対してアクセスしても動作を停止します。

DMAの使用例として、DMAの初期化プログラムと、V-RAMの内容を0FFHが来るまで転送するプログラムを示します。

リスト6-2 DMA初期化サブルーチン

```

DAMADD EQU      1F80H
DMAIN:  LD       BC, DMAADD
        LD       HL, DMIDT
DMAI1:  LD       A, (HL)
        CP       0FFH
        JR       Z, DMAI2
        OUT      (C), A
        INC      HL
        JR       DAMI1
DMAI2:  RET
;
DMIDT:  DB       83H, 00H, 14H, 14H, 80H
        DB       81H, 80H, 0C7H, 0CBH, 87H
        DB       0FFH
        ;
        END

```

DMAへコマンドとデータの送信

リスト6-3 V-RAMの内容サーチ・アンド・転送

```

DMAADD EQU      1F80H
VRSRC:  LD       BC, DMAADD
        LD       HL, VRSDT
VRSR1:  LD       A, (HL)
        CP       0FFH
        JR       Z, VRSR2
        OUT      (C), A
        INC      HL
        JR       VRSR1
VRSR2:  RET
;
VRSDT:  DB       83H, 7FH, 00H, 40H, 00H, 20H
        DB       5CH, 00H, 58H, 00H, 94H, 0FFH
        DB       0ADH, 00H, 80H, 92H, 0CFH, 87H
        DB       0FEH
        ;
        END

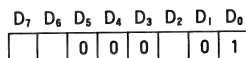
```

DMAへコマンド、データの送信

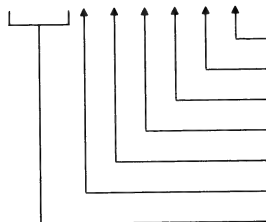
●読み出しレジスタ

DMAには、動作状態を知るために7個の読み出しレジスタがあります。読み出しレジスタの一覧を図6-6に示します。

読み出しレジスタ 0



ステータス・レジスタ



1 = DMA 動作中 (ロードコマンド (CFH) によりリセット)

0 = レディ・アクティブ

未使用

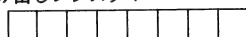
0 = 割り込み保留

0 = 一致成立

0 = エンド・オブ・ブロック → ステータスバイト初期再設定コマンド (8BH) によってリセット

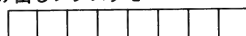
未使用

読み出しレジスタ 1



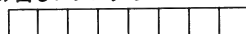
バイト・カウンタ (下位バイト)

読み出しレジスタ 2



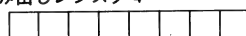
バイト・カウンタ (上位バイト)

読み出しレジスタ 3



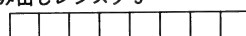
ポート A アドレス・カウンタ (下位バイト)

読み出しレジスタ 4



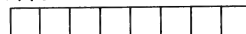
ポート A アドレス・カウンタ (上位バイト)

読み出しレジスタ 5



ポート B アドレス・カウンタ (下位バイト)

読み出しレジスタ 6



ポート B アドレス・カウンタ (上位バイト)

図6-6 Z80-DMA読み出しレジスタ

読み出しレジスタの内容を読む方法は2つあります。

(1) 読み出しステータスバイトコマンド

DMA に 0BFH (読み出しステータスバイトコマンド) を書き込みます (書き込みレジスタ 5 に書き込むことになる)。続いて DMA を IN 命令で読み出します。読み出せるのは、ステータスレジスタ (読み出しレジスタ 0) だけです。

(2) 読み出しシーケンスコマンド

このコマンドを使うと、希望のレジスタだけを読み出すことができます。

DMA に 0BBH (読み出しマスク指定コマンド) を書き込みます。続いて、読み出したいレジスタの該当のビットを 1 にした、マスクバイトを書き込みます (図 6-5 参照)。次に 0A7H (読み出しシーケンスコマンド) を書き込み、続いて IN 命令でレジスタを読み込みます。読み込まれるレジスタはマスクバイトで指定したレジスタで、順番はレジスタ 0 から 6 の方向です。

DMA が終了した後の読み出しレジスタの内容は、表 6-8 のようになっています。実際に転送されるバイト数や、DMA 終了後のカウンタの値は、動作クラス (転送, サーチ, 転送&サーチ) や動作モードによって少しずつ違うことに注意して下さい。

動作クラス	動作モード	実際に転送／サーチされるバイト数	バイトカウンターの内容	アドレスカウンターの内容(ソース)	アドレスカウンターの内容(デスティネーション)
転送		$x + 1$	x	$a \pm (x + 1)$	$a \pm x$
転送／サーチ		x	$x - 1$	$a \pm x$	$a \pm (x - 1)$
サーチ	バイト バースト 連続	x $x + 1$ $x + 1$	x $x + 1$ $x + 1$	$a \pm x$ $a \pm (x + 1)$ $a \pm (x + 1)$	

※ x は、WROに設定した転送ブロック長、またはサーチで一致した時のバイト数、 a は、アドレスカウンターの設定値である。

表6-8 DMAの読み出しレジスタの内容

リスト6-4 DMAのレジスタをすべて読み出して格納

```
DMAADD EQU      1F80H
DMARD:  LD        BC, DMAADD
        LD        HL, DMRDT
DMAR1:  LD        A, (HL)
        INC       HL
        CP        0FFH
        JR        Z, DMAR2
        INC       HL
        OUT       (C), A
        JR        DMAR1
DMAR2:  LD        D, 07H
        IN        A, (C)
        LD        (HL), A
        DEC       D
        INC       HL
        JR        NZ, DMAR2
        RET
;
DMRDT:  DB        83H, 0BBH, 00H, 0A7H, 0FFH
;
DS      7
;
END
```

DMAへコマンドの送信

DMAからデータの受信

6-4 CTC

X1turbo には、Z80-CTC(Counter／Timer Circuit)が内蔵されています。CTCはその名の通り、プログラムによってカウンターやタイマーとして使えるLSIです。X1turboでは、主にRS-232Cとマウスのクロック発生用として使用しています。余ったチャンネルを使用して、プログラムに、定期的に割り込みをかけたりすることもできます。

6-4-1 CTCの概要

CTCは4つのチャンネルからなっています。各チャンネルは、プログラム設定により、外部のクロック入力をカウントするカウンターモードか、内部クロック(4Mz)をカウントするタイマーモードのどちらかで動作します。どちらのモードでも、カウント終了時に割り込みをかけることができます。X1turboでは、CTCは、図6-7のように配線されており、従って各チャンネルは次のように使用します。

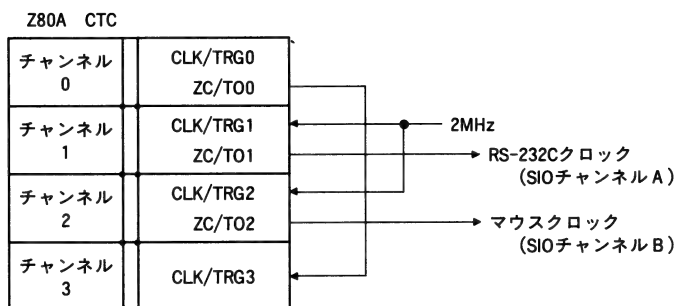


図6-7 CTCの配線

(1) チャンネル 0

このチャンネルの入力には何もつながっていないので、4Mz の内部クロックを使ったタイマーモードで使します。チャンネル 0 の出力パルスは、チャンネル 3 に接続されています。

(2) チャンネル 1

2Mz のクロックが入力されており、カウンターモードで使します。出力は Z80-SIO のチャンネル A に接続されており、RS-232C のボーレートが、CTC の設定によって決まります。

(3) チャンネル 2

チャンネル 1 と同様 2Mz のクロックが入力されています。出力は Z80-SIO のチャンネル B に接続されており、マウスのボーレートの作成に使します。

(4) チャンネル 3

チャンネル 0 の出力パルスが入力されており、チャンネル 0 だけでは作れない、長い周期を設定するときに使います。

6-4-2 CTC の使い方

X1turbo における CTC の I/O アドレスを次に示します。

チャンネル	I/O アドレス
0	1FA0H
1	1FA1H
2	1FA2H
3	1FA3H

表6-9 CTCのI/Oアドレス

CTC には、チャンネル制御レジスタと時間定数レジスタが、各チャンネルごとに 1 つずつあります。また、割り込みベクタレジスタが 1 つあります。チャンネル制御レジスタと割り込みベクタレジスタは、書き込まれたデータの最下位ビットが 0 か 1 かで判断されます。また、時間定数レジスタは、チャンネル制御レジスタのビット 2 を 1 にした後に書き込みます。割り込み制御レジスタは 1 つしかなく、どのチャンネルに書いても同じです。各レジスタの内容を図 6-8 に示します。

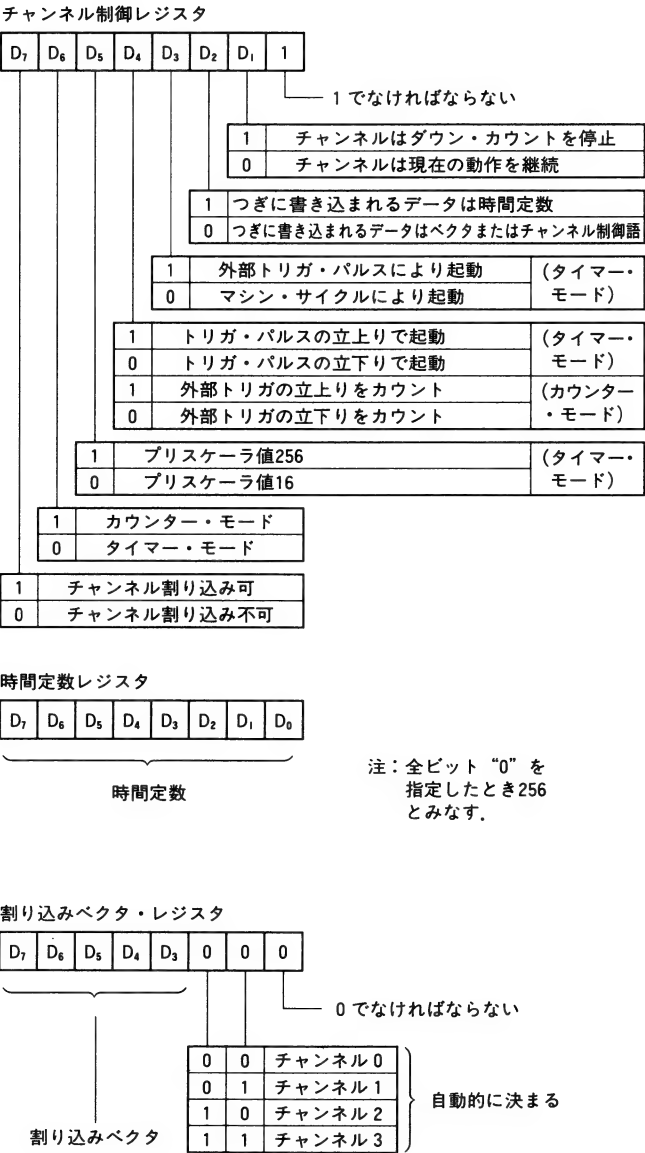


図6-8 Z80-CTCのレジスタ

6-5 キー入力

X1 シリーズでは、キー入力を割り込み処理によって行うことができます。
キー入力処理はサブ CPU が行っています。割り込みベクタの設定、キーデータの受け取り方などは、サブ CPU の章を参照して下さい。

第 7 章

フロッピーディスク

7-1 ディスクの物理フォーマット

X1 シリーズにおいては、表 7-1 に示す種類のコロッピーディスクがサポートされています。

内 容	記 録 方 式	記録内容	ディスク名
2D X1 フォーマット	両 面 倍密度記録	320K	3 インチ 5 インチFD
2DD X1 フォーマット	両面倍トラック 倍密度トラック	640K	
2HD X1 フォーマット	両面高密度 倍密度記録	1M	
*2HD 8 インチ 標準フォーマット	両 面 倍密度記録	1M	8 インチFD
2D X1 フォーマット	両 面 倍密度記録	1M	
*2D 8 インチ 標準フォーマット	両 面 倍密度記録	1M	
1S 8 インチ 標準フォーマット	片 面 単密度記録	240K	
X1 フォーマット	4 ヘッド 倍密度記録	10M	ハードディスク

*但しヘッド0、シリング0のみ1セクタ=128バイトの単密度記録

表7-1 ディスクの種類

7-1-1 物理構造とディスクマップ

ディスク間でデータの書き込みや読み出しを行う場合には、ディスク上のアドレス情報(ヘッド番号、トラック番号、セクタ番号等)をもとにして、データの格納場所を検索し実行します。このため、ディスク上にこのアドレスを割り付ける必要があります。このアドレスの割り付けのことをフォーマット(1次フォーマット)と呼びます。

市販の 3 インチまたは 5 インチのディスク(2D、2DD)はフォーマットされていないので、必ずフォーマットしなくてはなりません。 8 インチディスク及び 5 インチ 2HD のディスクは標準フォーマットがかけられており、必ずしもフォーマットは必要ではありません。また、全セクタとも256バイト/セクタ、倍密度記録の「X1 フォーマット」にフォーマットして使用することもできます。

7-2 BASIC のファイル管理

7-2-1 物理アドレスと論理アドレス

フロッピーディスクのある特定のセクタの指定は、ヘッド番号、シリンドラ番号、セクタ番号を組み合わせることによって行います。これを、そのセクタの物理アドレスと呼びます。

一方、X1 シリーズの BASIC では、各セクタに連続した通し番号(レコード番号)をつけて管理しています。これをセクタの論理アドレスと呼びます。HuBASIC でディスクを管理する場合は16レコード(256×16=4K バイト)を最小単位として扱います。これをクラスタと呼びます。

7-2-2 ディレクトリ

ディレクトリには、ファイル名、属性、ファイルを更新した時の SAVE 年月日時分、ファイル先頭クラスタ番号などが格納されています。これによってファイル名と実際にファイルが格納されている場所との対応がつけられます。1ファイルあたりのディレクトリ情報は32バイトで構成されています。

各タイプのディスクはともに階層ディレクトリ形式なので、データ領域内に下位ディレクトリが設定され、各下位ディレクトリはディスクタイプに関係なく1クラスタ分(4K バイト:128ファイル分)の領域を確保できます。最上位ディレクトリ(ルートディレクトリ)で管理できるファイル数はディスクタイプで異なり次に示す通りです。

ディスク	項 目	ルートディレクトリ 領域 レコード数	ファイル数
3 インチまたは 5 インチFD	2D	16	78
	2DD	16	158
	2HD	16	247
8 インチFD	2D	16	247
ハードディスク		16	2504

表7-2 ルートディレクトリで管理できるファイル数

階層ディレクトリでは、各ディレクトリ同志の関係は図のようなツリー構造になります。

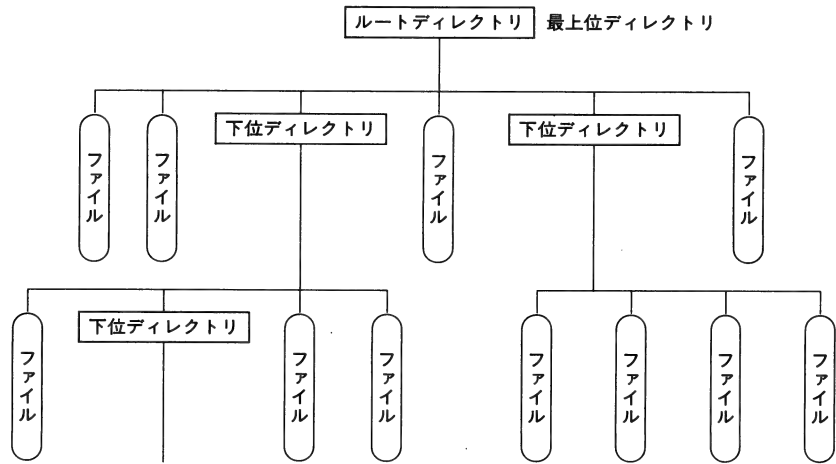


図7-1 ディレクトリのツリー構造

7-2-3 FAT

ファイル・アロケーション・テーブル(File Allocation Table 略して FAT)はファイルの格納状態を示します。ファイルが1つのクラスタに納まらない場合、残りを別のクラスタに書き込まなければなりません。次のクラスタが空いているとは限りません。そこで、どのクラスタが空いているか、また、このクラスタの次にどのクラスタに書き込んだか等の情報を記録したのがFATです。BASICはディレクトリとFATを使ってファイルを管理しているのです。FATの構成を図7-2に示します。

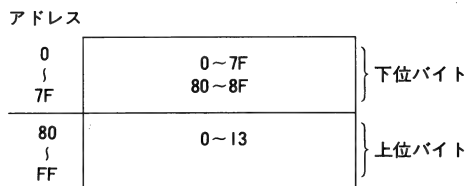


図7-2 FATの構成

1つのクラスタに対する情報は、FAT上では2バイトの数値で表わされますが、連続した2バイトではなく、図のようにFAT領域を128バイトずつに分割し、各々にデータの下位、上位バイトを書き込むようになっています。この256バイトの1レコードが、FATとして必要な容量(2バイト×最大ファイル数)になるまで続きます。FAT先頭にディスク管理用として、表7-3に示すバイト数が予約されています。このようにややこしい構造になっているのは、もともと5インチ2D(320Kバイト)のディスク用に設計された仕様を、ディスクの容量増加に伴って、無理に拡張したためであると思われます。

3インチまたは5インチFD2D, 2DD	2バイト
5インチFD2HD 8インチFD2D	3バイト
ハードディスク	4バイト

図7-3 ディスク管理用の予約バイト

FATの各クラスタに対する2バイトのデータの意味は、次の表に示す通りです。5インチ2Dを例にとると、そのクラスタが次のクラスタに続いている場合は、次のクラスタに対応する番号(02H~4FH)を、そのクラスタに続くクラスタがない場合には、クラスタ中の使用セクタ数に80Hを足した数(80H~8FH)を書き込むことを意味します。2Dディスク以外のディスクでは、この方法ではクラスタ番号が不足しますが、7FHの次は100Hというように、番号を不連続に割り振ることによってこれを解決しています。

項 目	3インチまたは5インチFD			8インチFD	ハード ディスク
	2D	2DD	2HD	2D	
ファイルがチェーンしている クラスタ	02~4F	02~7F 100~ 11F	03~7F 100~ 179	03~7F 100~ 179	04~7F 100~17F 1300~1353
ファイル終了クラスタ	80~8F	80~8F	80~8F	80~8F	80~8F
未使用クラスタ	0	0	0	0	0

表7-3 FATデータ

7-3 フロッピーディスクコントローラ(FDC)

7-3-1 FDC の概要

X1 シリーズでは FDC(フロッピーディスクコントローラ)として MB8877A を使用しています。FDC は FDD(フロッピーディスクドライブ)をコントロールするための LSI で、CPU からソフトウェアによってコントロールされます。おもな働きを次に示します。

1. データの変調と復調

CPU からのデータを変調してフロッピーディスクに書き込み、フロッピーディスクからのデータを復調して CPU に出力します。

2. FDD 駆動信号の発生と制御

ヘッドの移動など、FDD をコントロールします。

3. FDD 状態検出信号による制御

FDD の状態を検出して制御するとともに、CPU に通知します。

4. ホスト CPU とのインターフェイス

CPU からの FDD のコントロール、データのライト、および FDD から CPU へデータのリードを行います。

7-3-2 FDC レジスタと I/O ポート

FDC 内部には以下に示す種類のレジスタがあります。

1. コマンドレジスタ(CR)：書き込み専用 8 ビットレジスタ

FDC に実行させたい処理に応じたコマンドを CPU 側から書き込みます。なお、FDC がコマンドを実行している最中に次のコマンドを書き込んではいけません(フォースインタラプトコマンドを除く)。その場合の動作は保証されません。

2. ステータスレジスタ(STR)：読み出し専用 8 ビットレジスタ

FDC の内部状態、コマンド処理、FDD の状態などを表します。コマンドの種類によってそれぞれのビットが意味を持ちます。

3. データレジスタ(DR)：書き込み、読み出し可能な 8 ビットレジスタ

FDD からのデータを読む場合にはここにデータが置かれ、FDD にデータを書き込む時はこのレジスタに書き込まれたデータが FDD に出力されます。

4. トラックレジスタ(TR)：書き込み、読み出しが可能な 8 ビットレジスタ

通常は、ヘッドのあるトラック番号がセットされますが、コマンドによって、この値を変更することもしないことも可能です。

リードデータ・ライトデータコマンドでは、トラックレジスタの値と、ディスクから読み出された ID フィールド内のトラック番号を比較し、一致した場合にリード・ライトを実行します。

5. セクタレジスタ(SCR)：書き込み、読み出しが可能な 8 ビットレジスタ

リードデータ・ライトデータでは、セクタレジスタの値と、ディスクから読み出された ID フィールド内のセクタ番号を比較し、一致した場合にリード・ライトを実行します。

X1 シリーズの FDC の I/O アドレスは 0FF8H から 0FFFH までの 8 バイトで、ここには、FDC のレジスタの他に、FDD へのコントロール用 I/O ポートがあります。

0FFCH の出力ポートは、ドライブ番号、サイド、モータのオン・オフをコントロールします。ドライブの選択、サイドの選択、モータのオン・オフに関するレジスタはFDC 内にはありませんので、直接CPUから制御回路を通してFDDに与えます。なお、この情報をポートから読み取ることはできません。ドライブ番号セレクトはモータオン状態の時のみ有効です。

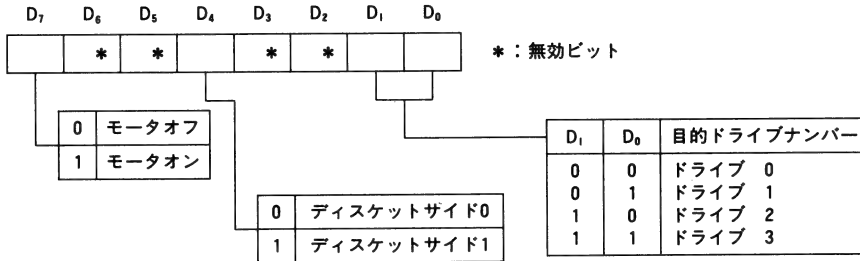


図7-4 0FFCHのデータビット内容

7-3-3 MB8877A のコマンド

FDC のコマンドは全部で11種類ありますが、これらは次の4種類に分類することができます。

タイプ	コマンド名称	動作
I	リストア シーク ステップ ステップイン ステップアウト	トラック0へ、ヘッドを移動する 所定のトラックへ、ヘッドを移動する ヘッドを1トラック移動する ヘッドを1トラック内側へ移動する ヘッドを1トラック外側へ移動する
II	リードデータ ライトデータ	ディスクのデータ（データフィールド）を読み込む ディスク（データフィールド）へデータを書き込む
III	リードアドレス リードトラック ライトトラック	ディスクIDフィールドを読む ディスクの1トラック分の全データを読み込む ディスクへ1トラック分の全データを書き込む
IV	フォースインターラプト	割り込みを発生させる

表7-4 コマンドタイプ分類

各コマンドは、タイプIVコマンドを除いて、前のコマンドが終了しないうちに書き込んではいけません。コマンドが実行中であるか終了したかを調べるためには、ステータスレジスタのビット0を見ます。これが1の時はコマンド実行中であることを表し、0の時はコマンドの実行が終了して、次のコマンドを書き込んでもよいことを表しています。

(1) タイプI コマンド

タイプI コマンドはヘッドの移動を行います。ヘッドの移動には、ディスク面の動径方向の移動と、垂直方向への移動があります。動径方向とはトラックからトラックへの移動のことで(シーク, リストア), 垂直方向とはヘッドをディスクに接触させるか(ロード), ディスクから離すか(アンロード)ということです。

コマンド名称	コマンドレジスタビット (MSB) (LSB)
リストア	0000h Vr ₁ r ₀
シーク	0001h Vr ₁ r ₀
ステップ	001uh Vr ₁ r ₀
ステップイン	010uh Vr ₁ r ₀
ステップアウト	011uh Vr ₁ r ₀

表7-5 タイプ I コマンド

・ステップレートフラグ(r₁ r₀)

ステップレートとはステップパルスの出力間隔のことで、シーク時のヘッドの移動速度を決定します。実際には(r₁ r₂)の他に FDC に与えられているクロック周波数と TEST 端子の状態に依存します。

TEST		“H”もしくは開放		“L”	
r ₁ r ₀	CLK	2 MHz	1 MHz	2 MHz	1 MHz
0	0	3msec	6msec	Approx 200μsec	Approx 400μsec
0	1	6msec	12msec		
1	0	10msec	20msec		
1	1	15msec	30msec		

表7-6 ステップレート

X1 シリーズでは TEST 端子は“H”，クロックは 2MHz が与えられていますので，表中の最左列だけが有効です。通常は，もっとも速い値 3msec を用います。

・ヘッドロードフラグ(h)

コマンドの実行開始時にヘッドをロードするか(h = 1)，あるいはアンロードするか(h = 0)を指示します。

・トラックレジスタ更新フラグ(u)

ヘッドの移動の際にトラックレジスタを更新するか(u = 1)，あるいは更新しないか(u = 0)を指示します。

・トラック照合フラグ(V)

ヘッド移動後，トラックレジスタの値と，FDD から読み取った ID フィールド内のトラック番号の照合を行うか(V = 1)，あるいは行わないか(V = 0)を指示します。

リストアコマンドは，トラック 0 へのシークと同じですが，シークコマンドが現在のトラック番号を与えなければならないのに対して，ヘッドがどこにあっても必ずトラック 0 に移動します。データリード、データライトなどが誤動作する場合には，いったんリストアを実行するとうまくいくことがあります。

ステップコマンド，ステップインコマンド，ステップアウトコマンドは 1 トラック分のシークですが，通常は使う必要がないでしょう。

(2) タイプII コマンド

タイプII コマンドは、ディスクのデータフィールドに対する書き込みと読み出しを実行します。リード・ライトの対象となるトラック番号とセクタ番号は、あらかじめトラックレジスタ、セクタレジスタに用意されていなければなりません。FDD とのデータの転送はソフトウェアでデータレジスタを通して1バイトずつ行います。

コマンド名称	コマンドレジスタビット	
	(MSB)	(LSB)
リードデータ		100m SEC 0
ライトデータ		101m SEC ₀

表7-7 タイプII コマンド

・マルチレコードフラグ(m)

連続するセクタでリード・ライトするか(m = 1)、1セクタだけでリード・ライトするか(m = 0)を指定します。ただしマルチレコードを指定できるのは、同一トラックの同一サイドのセクタに限ります。

・ディレイフラグ(E)

HLD 信号を“H”とした後、15msec 待ってから HLT をサンプリングするか(E = 1)、ただちに HLT をサンプリングするか(E = 0)を指示します。

・比較フラグ(C)

S フラグで指示したサイド番号と、ディスクから読み取った ID フィールド内のサイド番号が一致するかどうかの比較を行うか(C = 1)、行わないか(C = 0)を指示します。

・アドレスマークフラグ(a)

a = 0 の時、ライトデータ時にアドレスマークに (FB) H (Data Mark) を書きます。

a = 1 の時、データアドレスマークに (F8) H (Deleted Data Mark) を書きます。

(3) タイプIII コマンド

タイプIII コマンドは、ID フィールドのリードおよびトラック全体のリード・ライトを行います。これらのコマンドは現在ヘッドのあるトラックに対して処理が行われます。

コマンド名称	コマンドレジスタビット	
	(MSB)	(LSB)
リードアドレス		1100 0E00
リードトラック		1110 0E00
ライトトラック		1111 0E00

表7-8 タイプIII コマンド

・ディレイフラグ(E)

タイプII コマンドの場合と同じです。

リードアドレスコマンドは、最初に出会った ID フィールドの内容を読み出します。

リードトラックコマンドは、トラック内の全データを読み出します。

ライトトラックコマンドは、トラックに全データを書き出します。

これらのデータは1バイトずつソフトウェアで転送しますが、特にライトトラックコマンドではデータの転送が遅れると(00)Hが書き込まれたものと判断され、誤ったデータが書き込まれてしまいます。また、ライトトラックコマンドでは、ディスクのフォーマットのために0F5H から 0FEH の値がデータレジスタに書き込まれた場合、特別の処理を行います。

DRの内容	MFMの場合 (DDEN="L")	
	ディスクに書き込むデータ	DRの意味
00 } F4	00 } F4	データ } データ
F5	A1 *	IDAM, DAMの前提データ
F6	C2 *	IDMの前提データ
F7	CRC1 CRC2	内部で計算されたCRC 2 バイトを書く
F8	F8	データ (注1)*
F9 FA	F9 FA	データ データ
FB	FB	データ (注2)*
FC	FC	データ (注3)*
FD	FD	データ
FE	FE	データ (注4)*
FF	FF	データ

(注1) A1 *3バイトに続けて書かれた場合、デリーテッドデータマークとなる

(注2) A1 *3バイトに続けて書かれた場合、データマークとなる

(注3) C2 *3バイトに続けて書かれた場合、インデックスマークとなる

(注4) A1 *3バイトに続けて書かれた場合、IDアドレスマークとなる

表7-9 フォーマット用DR

(4) タイプIVコマンド

タイプIVコマンドは、フォースインターラプトコマンドの1つだけです。FDC がコマンドを実行中であるかないかにかかわらず、いつでもコマンドレジスタに書き込むことができます。フォースインターラプトコマンドが書き込まれた時、実行中のコマンドがあった場合は、そのコマンドは中止されます。この時、以下に示す条件でFDC の IRQ が "H" レベルになります。

コマンド名称	コマンドレジスタビット (MSB) (LSB)
フォース インターラプト	1101 <i>l</i> ₃ <i>l</i> ₂ <i>l</i> ₁ <i>l</i> ₀

I	IRQ発生条件
<i>l</i> ₀ =1	REDY入力の立ち上りでIRQ発生 (IRQ = "H")
<i>l</i> ₁ =1	REDY入力の立ち下りでIRQ発生
<i>l</i> ₂ =1	各インデックスパルス(IP)でIRQ発生
<i>l</i> ₃ =1	無条件でただちにIRQ発生

表7-10 タイプIVコマンド

(5) ステータス

FDDの状態やFDCのコマンド実行結果、コマンド実行中の状態はステータスレジスタに表れます。ステータスレジスタはコマンドのタイプによって各ビットが違う意味を持ちます。コマンド実行開始時に所定のビットがプリセットされ、コマンド実行終了直前に内容が確定します。

なおFDCはコマンドの終了、中止をIRQで通知しますが、X1シリーズではIRQを使用しておらず、ステータスレジスタの所定のビットを調べることによって判断しなければなりません。DRQは、データレジスタのリードまたはライトによってリセットされます。

コマンド	ステータス	意味
タイプ I コマンド	NOT-READY (STR7)	NOT-READY=1でディスクドライブが動作可能状態でない事を示します。READYとMRの論理和です。
	WRITE-PROTECT (STR6)	WRITE-PROTECT=1でディスクへの書き込みが禁止されている事を示します。WPRT入力 of 反転コピー。
	HEAD-ENGAGED (STR5)	HEAD-ENGAGED=1でヘッドがメディアに接してすることを示しています。HLDとHLTの論理和。
	SEEK-ERROR (STR4)	SEEK-ERROR=1でベリファイ動作が成功しなかったことを示します。
	CRC-ERROR (STR3)	CRC-ERROR=1でIDフィールド読み出しエラーがあったことを示します。
	TRACK00 (STR2)	TRACK00=1でディスクのヘッドがトラック0の上にあることを示します。TRACK00はTR00の反転コピーです。
	INDEX (STR1)	INDEX=1でインデックスホールの検出を示します。IP入力の反転コピー。
	BUSY (STR0)	BUSY=1でFDCがコマンド動作中である事を示します。
タイプ II / III コマンド	NOT-READY (STR7)	NOT-READY=1でディスクドライブが動作可能状態でない事を示します。READYとMRの論理和です。
	WRITE-PROTECT (STR6)	WRITE-PROTECT=1でディスクへの書き込みが禁止されている事を示します。WPRT入力 of 反転コピー。
	RECORD-TYPE2/ WRITEFAULT (STR5)	リード動作の時、RECORD TYPEの表示としてDMAがDDMのときにセットされます。 ライト動作の時、書き込み動作が打ち切られた事を示します。WFの反転コピーです。
	RECORD-NOT-FOUND (STR4)	RECORD-NOT-FOUND=1で指定されたトラック番号、サイド番号、セクタ番号の持ちながら、正しく読み出せたIDフィールドがなかったことを示します。
	CRC-ERROR (STR3)	CRC-ERROR=1で読み出しエラーがあったことを示します。
	LOST-DATA (STR2)	LOST-DATA=1で所用時間内にDRの読み出し、書き込みが行われなかったデータがあったことを示します。
	DATA-REQUEST (STR1)	DATA-REQUEST=1で、FDCがCPUに対してDRの読み出しあるいは書き込みを要求していることを示しています。DRQのコピーです。
	BUSY (STR0)	BUSY=1でFDCがコマンド動作中である事を示します。

表7-11 ステータスレジスタ

7-4 FDD のアクセス

7-4-1 MB8877A のアクセス

フロッピーをアクセスする時は、前もってドライブ番号、サイド番号、モータ制御(オンでなければならない)を0FFCHの出力ポートに出力し、次いでFDCにコマンドを与えます。FDCはコマンドを解析して処理を開始します。

X1 では MFM 方式の 320K バイトしかサポートしていませんでしたが、X1turbo では、記録方式では FM, MFM の 2 つを選択でき、記憶容量では 320K バイト, 640K バイト, 1M バイト(フォーマット時)の 3 つを選択できるようになっています。これらを切り換えるには所定のポートを IN 命令で読み出します。従って、FDC へのコマンドは次のような手順で実行します。

リスト7-1 FDCへのコマンドヘッダープログラム

```

LD      BC, 0FFCH
LD      A, 80H
OUT     (C), A
IN      A, (C)
LD      C, 0FBH .....FM方式で記録
LD      A, 目的のトラック番号
OUT     (C), A
LD      C, 0F9H
LD      A, 現在のトラック番号
OUT     (C), A
LD      C, 0F8H
LD      A, コマンド
OUT     (C), A

```

FDCにドライブ番号, サイド, モニタのオンオフを送信

リストア, シークのプログラム例を次に示します。

リスト7-2 リストア

```

                ドライブ番号
RESTR: LD      BC, 0FFCH
LD      A, 0+80H
OR      80H
OUT     (C), A
LD      C, 0F8H
LD      A, 00H
OUT     (C), A
REST1: IN      A, (C)
AND     01H
JR      NZ, REST1
RET
;
END

```

FDCにDドライブ番号を知らせる

FDCコマンドレジスタへリストアを送信

コマンド終了まで待つ

リスト7-3 シーク

```

TRACK EQU      01H
RESTR: LD      BC, 0FFCH
LD      A, 0+80H
OUT     (C), A
LD      C, 0FBH
LD      A, TRACK
OUT     (C), A
LD      C, 0F8H
LD      A, 10H
OUT     (C), A
REST1: IN      A, (C)
AND     01H
JR      NZ, REST1
RET
;
END

```

FDCにドライブ番号を送信

FDCデータレジスタに目的のトラック番号を入れる

FDCコマンドレジスタにシークを送信

コマンド終了まで待つ

7-4-2 リード・ライトプログラム

Xlturbo は DMA を搭載しているため、DMA で直接ディスクから RAM ヘメモリを転送することが可能です。

ディスクとのデータのリード・ライトは256バイト(1セクタ)を単位として行われますので、ソフトウェアでも、1セクタのリード・ライトルーチンを用意して必要なだけのセクタ数のデータをリード・ライトすると便利です。

(1) DMA 未使用時の手順(X1 の場合)

1. FDD がリード・ライト可能であるかどうかチェックします。
 - ・ドライブ、サイドをセレクト。モータオン。
 - ・FDD が READY になるのを待ちます。
2. リード・ライトする目的トラックへヘッドを移動します。
 - ・(必要ならば)記録方式を設定します。
 - ・(必要ならば)記録容量を設定します。
 - ・目的トラック番号をデータレジスタに書き込みます。
 - ・現在ヘッドがあるトラックの番号をトラックレジスタに書き込みます。
 - ・シークコマンドをコマンドレジスタに書き込みます。
 - ・エラーが発生した場合は5回程度リトライします。
3. 目的セクタとの間でデータを転送します。
 - ・メモリ上のバッファアドレスを設定します。
 - ・目的セクタ番号をセクタレジスタに書き込みます。
 - ・リード・ライトコマンドをコマンドレジスタに書き込みます。
 - ・データレジスタを通してデータを1バイトずつ転送します。
 - ・エラーが発生した場合は5回程度リトライします。

データ転送はデータレジスタを通して1バイトずつ行いますが、1バイトの転送は16 μ sec以内に完了しなければなりません。特にライトコマンドでこれを怠りますと、(00)Hが書き込まれたものと判断されてしまいます。

リード・ライトの終了はステータスレジスタの BUSY(ビット0)が0になることで分かります。

(2) DMA 使用時の手順

1. DMA 初期設定

リード動作時

- ・ポートA → ポートB転送モードに設定します。
- ・ポートAのアドレスとして、FDC のデータレジスタのアドレスに設定します。
- ・ポートAを、I/O アドレスに設定します。(WR1)
- ・ポートBを、メモリ、アドレスインクリメントに設定します。(WR2)
- ・ブロックがエンドでストップ、RDY をアクティブローとします。
- ・DMA をイネーブルにします。(WR6)

ライト動作時

- ・ポートB → ポートA転送モードに設定します。(WR0)

- ・ポートAの開始アドレスを、RAMの転送開始アドレスに設定します。
 - ・ポートAを、メモリ、アドレスインクリメントに設定します。(WR1)
 - ・ポートBを、I/O、アドレスに固定に設定します。(WR2)
 - ・動作モードをバイトモードに設定し、ポートBの開始アドレスを設定します。
 - ・両ポートの開始アドレスをロード、バイトカウンタをクリアします。
 - ・ポートA → ポートB転送モードに設定します。
 - ・DMAをイネーブルにします。
2. FDDがリード・ライト可能であるかどうかチェックします。
- ・ドライブ、サイドをセレクト。モータオン。
 - ・FDDがREADYになるのを待ちます。
3. リード・ライトする目的トラックへヘッドを移動します。
- ・(必要ならば)記録方式を設定します。
 - ・(必要ならば)記録容量を設定します。
 - ・目的トラック番号をデータレジスタに書き込みます。
 - ・現在ヘッドがあるトラックの番号をトラックレジスタに書き込みます。
 - ・シークコマンドをコマンドレジスタに書き込みます。
 - ・エラーが発生した場合は5回程度リトライします。
4. リード・ライト
- ・目的セクタ番号をセクタレジスタへ書き込みます。
 - ・リード・ライトコマンドをコマンドレジスタへ格納します。
 - ・ステータスレジスタのBUSYが0になるまでデータレジスタからRAMへ転送します。

第 8 章

サウンド機能

8-1 PSG

8-1-1 PSG (AY-3-8910) について

X1 シリーズでは PSG (Programmable Sound Generator) として AY-3-8910 を搭載しています。この PSG は 8 オクターブ、3 重和音発声機能を持つほか、2 つの汎用 I/O ポート (このポートは、ジョイスティック用として使用されています) を持っています。この PSG の構成は次の通りです。

- (1) A, B の 2 つの I/O ポート
- (2) 3 チャンネルのトーンジェネレータ
- (3) 1 チャンネルのノイズジェネレータ
- (4) 1 チャンネルのエンベロープジェネレータ

このトーンジェネレータは矩形波音源からなっていて、FM 音源の主役となるものです。一方、ノイズ、エンベロープはこの主役の引きたて役となります。ノイズは効果音を作る時に用います。また、エンベロープとは音量の時間的変化のことで、ピアノやオルガンなどに特有の音も、このエンベロープを変えることによって発生させることができます。ノイズとうまく組み合わせれば残響効果も楽しめます。

8-1-2 PSG の原理

PSG の内部構造を図 8-1 に示します。

この PSG は内部に 16 個のレジスタを持ちます。各レジスタの機能を以下に示します。

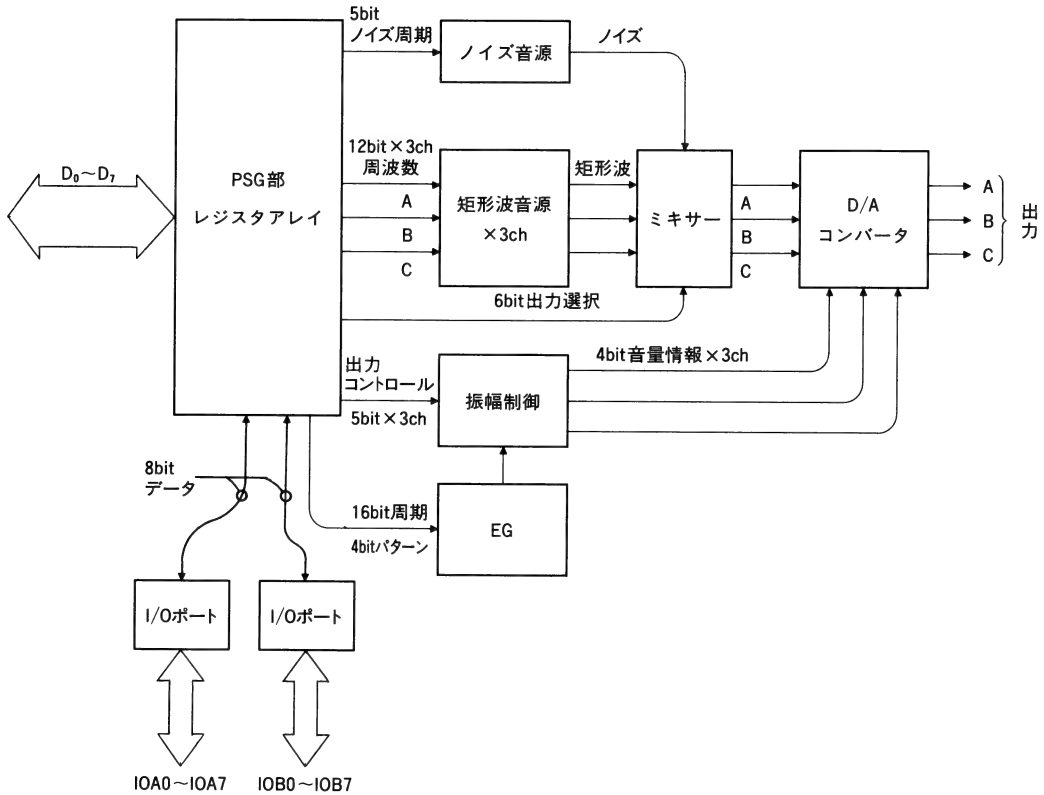


図8-1 PSGの内部構造

レジスタ	機 能	7	6	5	4	3	2	1	¥	備 考	
0	チャンネルA周波数8ビット微調整	00H~FFH									
1	チャンネルA周波数4ビット粗調整					00H~0FH					
2	チャンネルB周波数8ビット微調整	00H~FFH									
3	チャンネルB周波数4ビット粗調整					00H~0FH					
4	チャンネルC周波数8ビット微調整	00H~FFH									
5	チャンネルC周波数4ビット粗調整					00H~0FH					
6	ノイズ周波数5ビット設定					00H~1FH					
7	各チャンネル音量ON/OFF	IN/OUT		ノイズ			トーン			1 OFF 0 ON	
	ジョイスティックポート入出力スイッチ	B	A	C	B	A	C	B	A	1 OUT 0 IN	
8	チャンネルA 音量設定/エンベロープ				M	L ₃	L ₂	L ₁	L ₀	ビット4で 切り換え	
9	チャンネルB 音量設定/エンベロープ				M	L ₃	L ₂	L ₁	L ₀		
10	チャンネルC 音量設定/エンベロープ				M	L ₃	L ₂	L ₁	L ₀		
11	エンベロープ周期下位8ビット	00H~FFH									
12	エンベロープ周期上位8ビット	00H~FFH									
13	エンベロープ形状					E ₃	E ₂	E ₁	E ₀	0~15	
14	ジョイスティックI/OポートA	00H~FFH								8ビット パラレル データ	
15	ジョイスティックI/OポートB	00H~FFH									

表8-1 PSG(AY-3-8910)の各レジスタの機能

8-1-3 レジスタアクセス

PSG のレジスタへのアクセスはシステム I/O ポートによって行います。まず I/O ポート (1C ** H) に設定したいレジスタの番号を16進数 (00H-0FH) で書き込みます。次に、そのレジスタのリード、ライトを行います。音を発生させる手順は、トーン周波数 (R0~R5)、音量 (R8, R9, R10) を設定した後チャンネルスイッチ (R7) のビットを ON (0) にすることによって行います。

システムI/Oポート	IN/OUT	機 能	BDIR	BC1
——	——	PSG ノンアクティブ	0	0
(1B **)H	IN	PSGからの読み出し	0	1
	OUT	PSGへの書き込み	1	0
(1C **)H	OUT	レジスタを指定	1	1

* 印は無効ディジット

カスタム LSI のゲートにより、システム I/O ポート、及び IN/OUT から自動的に PSG の BDIR 端子、BC1 端子の値が決まります。

表8-2

8-1-4 トーン周波数設定 (R0~R5)

PSG には、3 つのトーンジェネレータが搭載されています。各々のトーンジェネレータはたがいに影響し合うことはありません。周波数の設定は粗調整レジスタ (R1, R3, R5) の下位 4 ビットと、微調整レジスタ (R0, R2, R4) の 8 ビットによって行われます。

チャンネル	粗調整レジスタ	微調整レジスタ
A	R ₁	R ₀
B	R ₃	R ₂
C	R ₅	R ₄

ビット構成	
	12ビット トーン周波数
	最小値: 000000000001 (1) 最大値: 111111111111 (4,095 ₁₀)

(a) トーン周波数レジスタの内容と構成

音階	音階	平均律周波数Hz	16進データ
ド	C	32.703	0 E E F
ド #	C #	34.648	0 E 1 A
レ	D	36.708	0 D 4 D
レ #	D #	38.891	0 C 9 0
ミ	E	41.203	0 B F 2
ファ	F	43.654	0 B 2 F
ファ #	F #	46.249	0 A 9 1
ソ	G	48.999	0 9 F 5
ソ #	G #	51.913	0 9 6 A
ラ	A	55.000	0 8 E 1
ラ #	A #	58.270	0 8 6 3
シ	B	61.735	0 7 E 9

(b) 音階設定周波数データ

表8-3 トーン周波数レジスタの内容と構成

このトーンジェネレータは12ビットの分周器として働き、125kHz を分周します。12ビットで表せる数は0 から4095ですので125kHz から30.5Hz まで、30.5Hz ステップの周波数が設定できます。

出力周波数に対する設定値の決め方

出力したい周波数を f (kHz) とすると

$$TP=125kHz/f$$
$$CT+FT/256=TP/256$$

TP : 分周比

FT : 微調整(R0, R2, R4)

CT : 粗調整(R1, R3, R5)

ただし、 $FT \leq 255$ とし、CT, FT とともに整数とします。もし、TP が整数にならないときは四捨五入した数で妥協します。

8-1-5 チャンネル音量設定(R8, R9, R10)

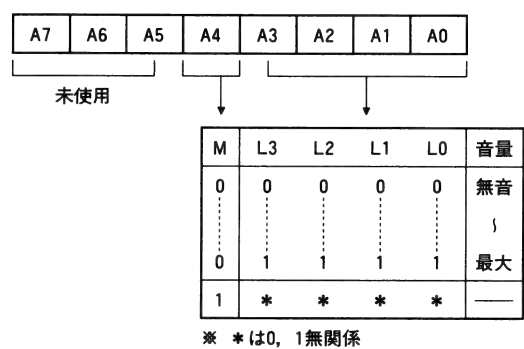


図8-2 チャンネル音量設定レジスタの内容と構成

チャンネルA, B, Cのトーンジェネレータにおける、音量設定モードとエンベロープモードの切り換えはレジスタ R8, R9, R10 のビット 4 によって行います。ビット 4 を 0 にすると音量設定が有効になり、16段階の音量設定ができます。1 にするとエンベロープモードになるので音量設定は無効になります。音を消すには消したいチャンネルの音量設定レジスタの内容を 0 にするか、チャンネルスイッチ(R7)を OFF(1)にします。

8-1-6 チャンネルスイッチ(R7)

チャンネルスイッチ(R7)は各チャンネルのトーン ON/OFF, ノイズの混合, 汎用 I/O ポートの入出力の切り換えを行うレジスタです。ただし、X1 シリーズではこの I/O ポートをジョイスティック用に使用していますので、単なる入力ポートとしてのみ使用することになります。

チャンネルスイッチ(R7)をアクセスする場合は、まず R7 の内容を読みだし、AND, OR 命令などで目的のビットのみをアクセスするようにします。つまり、OFF になっていたチャンネルを ON にするには、読みだした R7 のビットパターンに目的のビットだけが 0 で他のビットが 1 のビットパターンと AND をとり、得られた結果を R7 に書き込みます。逆に ON になっていたチャンネル OFF にするときは、目的のビットだけが 1 で、他のビットが 0 のビットパターンと OR をとります。

B7	B6	B5	B4	B3	B2	B1	B0

I / O ポート ノイズ トーン : 制御対象

ビット	説 明					
B2～B0	出力させたいトーンチャンネルを選択します。					
	B2	B1	B0	トーン出力チャンネル		
	0	0	0	C	B	A
	0	0	1	C	B	—
	0	1	0	C	—	A
	0	1	1	C	—	—
	1	0	0	—	B	A
	1	0	1	—	B	—
	1	1	0	—	—	A
	1	1	1	—	—	—

—印：チャンネルOFF
 他：チャンネルON

B5～B3	出力させたいノイズチャンネルを選択します。					
	B5	B4	B3	ノイズ出力チャンネル		
	0	0	0	C	B	A
	0	0	1	C	B	—
	0	1	0	C	—	A
	0	1	1	C	—	—
	1	0	0	—	B	A
	1	0	1	—	B	—
	1	1	0	—	—	A
	1	1	1	—	—	—

B7, B6	ジョイスティック端子2ポートの出力を指定します。					
	B5	B4	入出力制御			
	0	0	ジョイスティック端子(JS)1・2共に入力			
	0	1	JS2：入力, JS1：出力			
	1	0	JS2：出力, JS1：入力			
	1	1	JS1・2 共に出力			

表8-4 チャンネルスイッチレジスタの内容と構成

8-1-7 ノイズ周波数設定(R6)

PSG のノイズジェネレータはサンプル&ホールドという方式でノイズを発生させています。ここでいうノイズ周波数とはノイズが持つ周波数帯域幅のことです。ノイズ周波数を低く設定すると「ザー」というピンクノイズ、高く設定すると「シャー」というホワイトノイズを生じます。ノイズ周波数の設定は、レジスタ6の下位5ビットによって行われます。

N7	N6	N5	N4	N3	N2	N1	N0
未使用			ノイズ周波数値				

図8-3 ノイズ周波数設定レジスタの内容と機能

ノイズはA, B, Cのいずれかのトーンにミキシングされ出力されます。ノイズ音量だけを設定することはできず、ミキシングされたトーンとともに音量設定されることになります。

ノイズ周波数の設定法

出力したいノイズの平均周波数を f (kHz) とすると

$$R6 = 125\text{kHz} / f$$

8-1-8 エンベロープ周期設定レジスタ (R11, R12)

エンベロープモードにするにはエンベロープを付けたいチャンネルの音量設定レジスタのビット 4 を 1 にします。これにより音量設定は無効になりエンベロープモードになります。

エンベロープ周期は、粗調整レジスタ R12 の 8 ビット、微調整レジスタ R11 の 8 ビットによっておこないます。

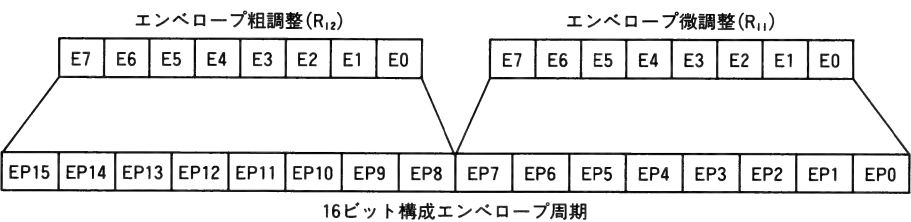


図8-4 エンベロープ周期設定レジスタのビット内容

R11, R12 は合わせて16ビット分周器として働きます。

エンベロープ周期の決め方

エンベロープ周波数を f (kHz) とすると

$$EP = 125\text{kHz} / f$$

EP : 分周比

$$R12 + R11 / 256 = EP / 256$$

R12 : 粗調整

R11 : 微調整

音量は最大から無音まで16段階に分けられ、PSG のカウントアップ／ダウンに合わせて段階的に変化します。

8-1-9 エンベロープ形状設定レジスタ (R13)

エンベロープ形状はレジスタ13の下位 4 ビットで設定します。

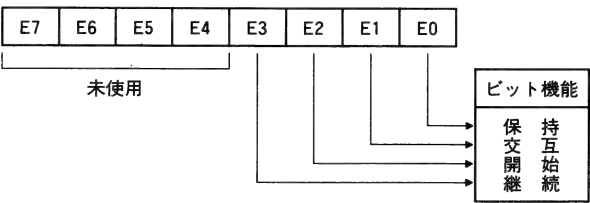


図8-5 エンベロープ形状設定レジスタのビット内容

各ビットの機能を次に示します。

(1) ビット 0 : 保持(Hold)

1 のとき、カウントアップ/ダウンモードにより一周期目のエンベロープカウンタの最終値 (0000 または 1111) における音量を以後ずっと保持します。

(2) ビット 1 : 交互(Alternate)

1 のとき、カウントアップの周期とカウントダウンの周期を交互に繰り返します。一周期目のカウントアップ/ダウンは開始ビットによります。

(3) ビット 2 : 開始(Attack)

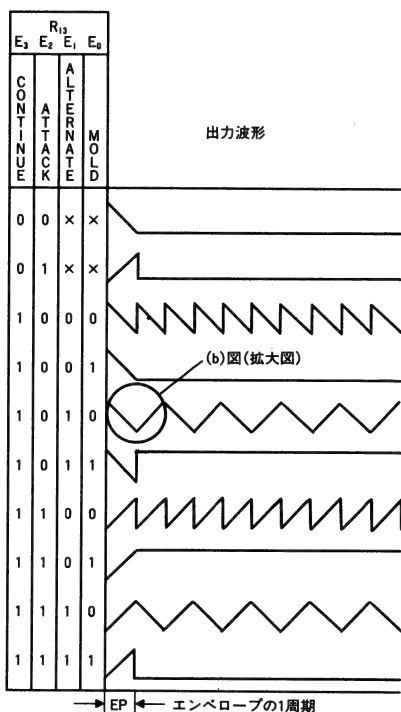
1 周期目のカウントアップ/ダウンを設定します。1 のとき、エンベロープカウンタは 0000 から 1111 までカウントアップし、0 のとき、逆に 1111 から 0000 までカウントダウンします。

(4) ビット 3 : 継続(Continue)

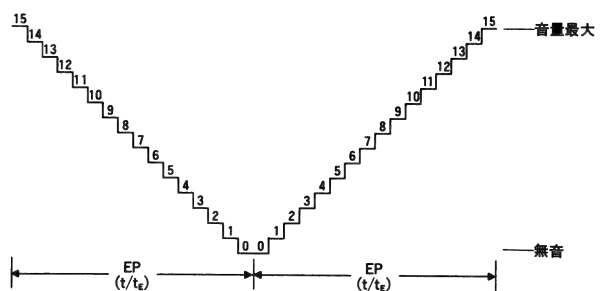
1 のとき、サイクルパターンは保持ビットの内容に依存します。0 のとき、一周期を実行した後、0000 値にリセットされそのとき音量に戻り、再びカウントを始めます。交互ビットが 1 のとき、この継続ビットは無効となります。また継続ビットが 0 のとき、保持ビット、交互ビットにかかわらず、一周期目を実行後、0000 値に置ける音量を以後ずっと保持します。(このとき保持ビットはカウンタの最終値、継続ビットはカウンタの初期値であることに注意してください)

なお、保持ビットと交互ビットを同時に 1 としたときはエンベロープカウンタは保持状態に戻る前に初期カウンタ値にリセットされます。

以上の 4 つのビットにより 10 通りのエンベロープの形状を設定できます。



(a) 各ビットと出力波形の対応



(b) エンベロープ波形拡大図

図8-6 エンベロープ形状の設定

8-1-10 ジョイスティックポートアクセス

PSGのレジスタ14, 15は, A, B 2つの8ビット汎用I/Oポートとして使用しています。X1シリーズではジョイスティック用入力ポートとして使用しています。このI/Oポートはサウンド機能と全く独立したもので, I/Oポートの使用により, 音が影響を受けることはありません。

8-2 FM 音源

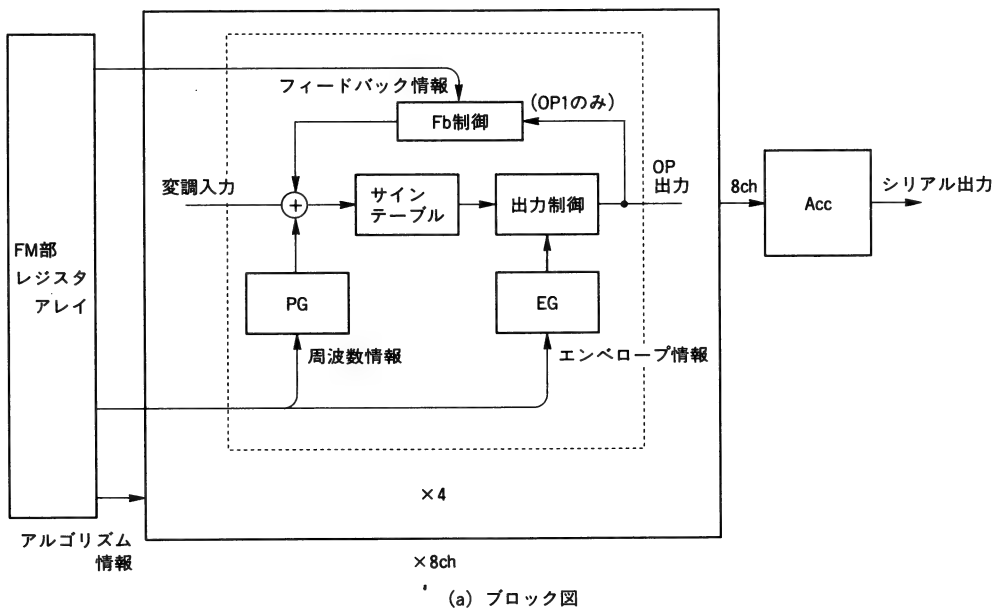
FM音源とは, いろいろな音源同志の波形の加算はもちろん, 一つの音源の出力で他の音源にFM変調(周波数変調)をかけることのできる音源です。これによりさまざまな音色を出すことができます。PSGでは簡単な矩形波とノイズしか出せないため音色の表現ができない点がFM音源との根本的な違いです。またFM音源では各音源(オペレータ)ごとにエンベロープジェネレータがついていますのでPSGと違って音量の変化を自由にできます。

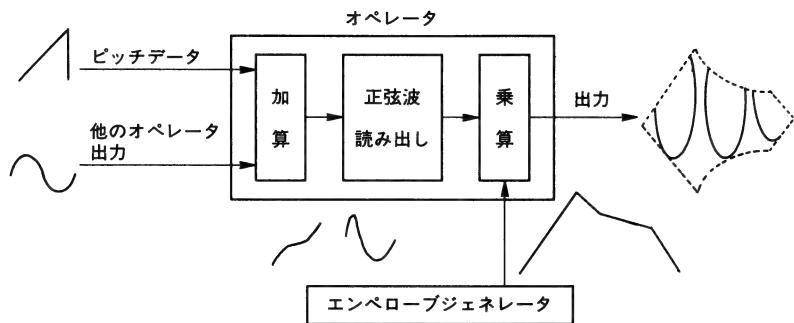
X1シリーズにはFM音源ボード(CZ-8BS1)が別売されています。また, X1turbo Zではこれと同等のFM音源が標準装備されています。

8-2-1 FM音源の原理

CZ-8BS1にはYM2151(OPM; FM OPERATOR TYPE-M)が搭載されていますこのYM2151は, 専用D/AコンバータであるYM3012と組み合わせることにより8音, R(right)/L(left)のステレオオーディオ信号が得られます。X1turbo Zでは, サウンドジェネレータとしてPSG(YM2149)も搭載されていますので, サウンド出力はFM音源とPSGがミキシングされた音が出力されます。他のX1シリーズではFM音源とPSGは全く独立に音を出力します。

FM音源の基本的な構成を下図に示します。





(b) 基本的な構成

図8-7 FM音源の基本的な構成

(1) オペレータ

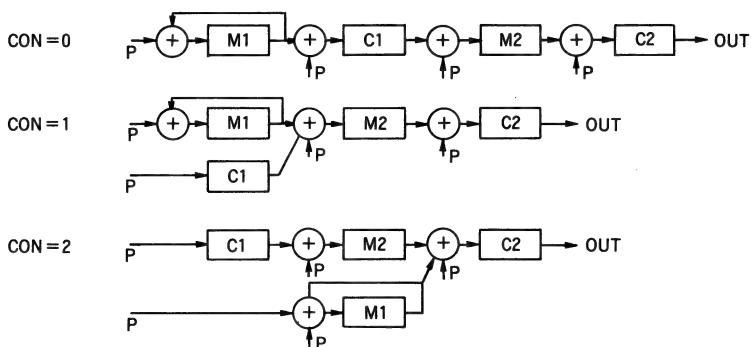
オペレータとは、自由に周波数と振幅を設定できる正弦波発振器のことです。このFM音源には、1つのチャンネルにつき4個のオペレータを持ち、従来のPSGと同時に出力するミキシングも可能です。オペレータの重要な機能はオペレータ(1)の出力を他のオペレータ(2)に変調信号として加えると、オペレータ(2)の出力はFM変調波となることです。オペレータ(2)中には“正弦波読み出し部”と呼ばれる部分があり、正弦波のデータがここにセットされていて、これをピッチデータとオペレータ(1)の出力を加算したものに見合う速度で読み出せば、FM変調波を作り出すことができます。この場合、オペレータ(1)はモジュレータ、オペレータ(2)はキャリアと呼ばれます。なお、4個のオペレータはすべて同等でありモジュレータ、キャリアというのは役割上の分類であることに注意してください。

(2) アルゴリズム

4個のオペレータの組合せ方をアルゴリズムといいます。このFM音源では全部で8種類のアルゴリズムを設定できます。まず、キャリアで音量をコントロールし、モジュレータによってキャリアに特性を出し音色をコントロールします。

M: モジュレータ

C: キャリア



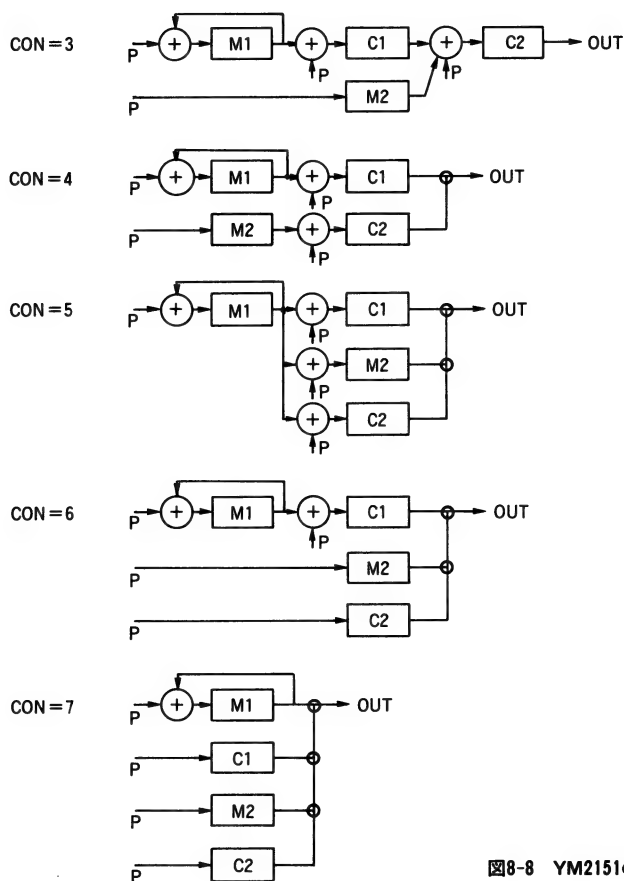


図8-8 YM2151の持つアルゴリズム

複数のモジュレータで1つのキャリアに変調をかける場合は同時に変調するよりは、FM 変調波でさらに FM 変調をかける方が、より多く倍音を生じ明るい音になります

8-2-2 FM 音源のアクセス

FM 音源は PSG に比べてはるかに多くのレジスタがありますので慣れるまではかなり大変です。FM音源のアクセスは、まず、レジスタ(00H~FFH)の設定から始めます。265個のレジスタのうち、どのレジスタをアクセスするかの設定はユーザー I/O ポート 700H(アドレスポート)で行い、データの書き込みは 701H(データポート)で行います。例えば

```
レジスタの設定: LD    BC, 700H ;
                  LD    A, レジスタの番号(** H) ;
                  OUT   (C), A ;
データ・ライト: LD    BC, 701H ;
                  LD    A, データ ;
                  OUT   (C), A ;
```

のように行います。

8-2-3 FM音源のレジスタ

FM音源の各レジスタ名と機能は以下の通りです。

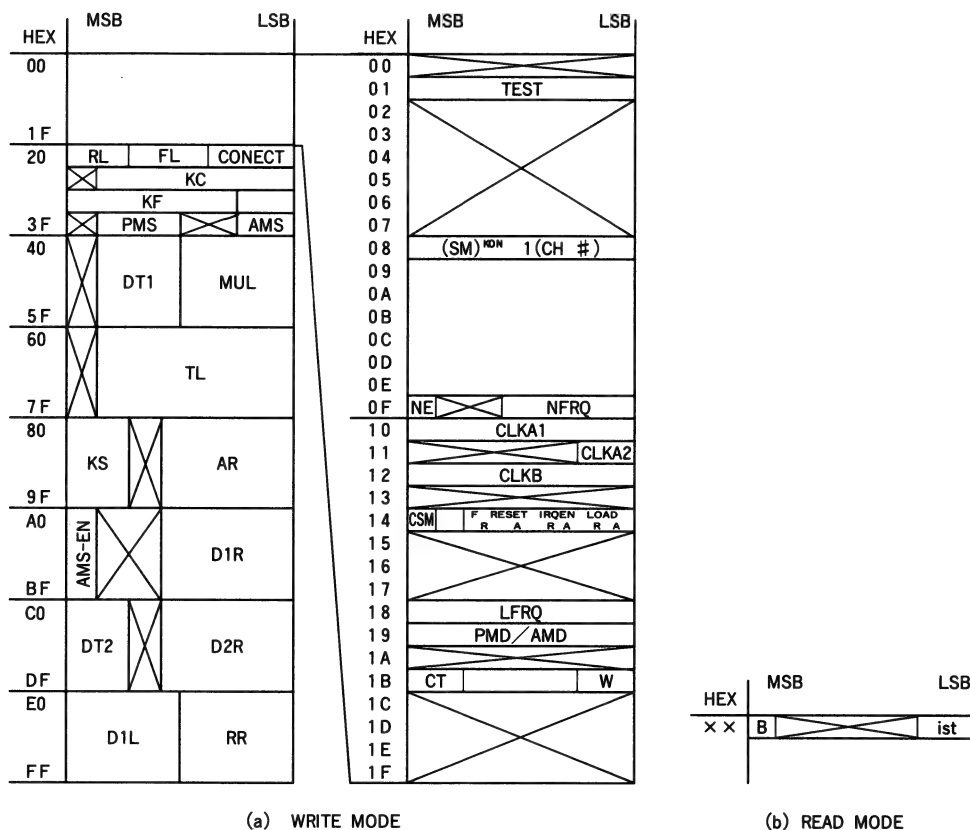


図8-9 アドレスマップ

(A)ライトモード(カッコ内は、レジスタ番号を示します)

(1) TEST(01H)

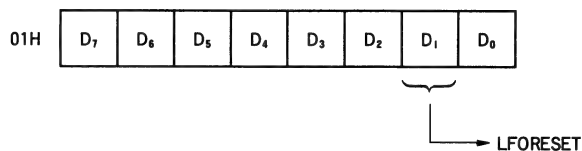


図8-10

YM-2151のテスト用レジスタで、ビット1以外は必ず0にしておきます。ビット1はLFOリセット用に使われていて、キーオンするときに一度1を書き込み、その後また0に戻すとLFOがリセットされ、初めから再スタートします。これにより各変調のキーオンのときの同期ができます。

(2) LFO : LOW FREQUENCY OSCILLATOR

LFO の低周波出力は、オペレータの音量や音程に作用して、ビブラートなどのさまざまな効果を作り出すことができます。ただし、LFO の効果が有効なのはキーオン～キーオフの期間内であることに注意して下さい。なお、この LFO は、約 53Hz から約 0.0008 Hz までの周波数が出せます。

● W : WAVE FORM (1BH)

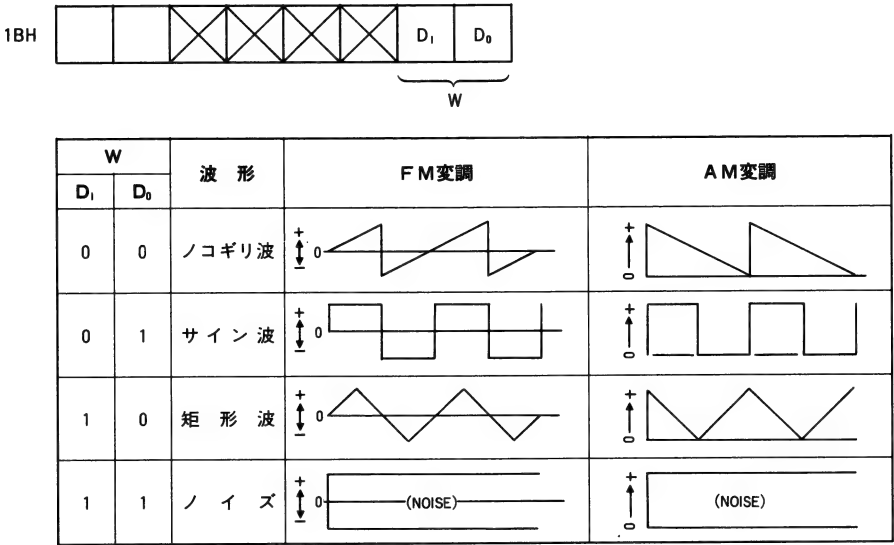


図8-11 出力できる波形

LFO は、ノコギリ波、サイン波、矩形波、ノイズの中から一つを選び出力でき、これに FM 変調と AM 変調(振幅変調)をかけることができます。この時、出力レベルは FM 変調、AM 変調とで別々に設定できます。

● CT : CONTROL OUTPUT (1BH)

使用していません。

● LFRQ : LOW FREQUENCY (18H)

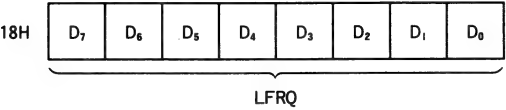


図8-12

8 ビットにより LFO の発振周波数を設定します(表 8-5)。

DATA (HEX)	FREQ. (Hz)	DATA (HEX)	FREQ. (Hz)	DATA (HEX)	FREQ. (Hz)	DATA (HEX)	FREQ. (Hz)
FF	52.9127	BF	3.3070	7F	0.2067	3F	0.0129
FE	51.2058	BE	3.2004	7E	0.2000	3E	0.0125
FD	49.4989	BD	3.0937	7D	0.1934	3D	0.0121
FC	47.7921	BC	2.9870	7C	0.1867	3C	0.0117
FB	46.0852	BB	2.8803	7B	0.1800	3B	0.0113
FA	44.3784	BA	2.7736	7A	0.1734	3A	0.0108
F9	42.6715	B9	2.6670	79	0.1667	39	0.0104
F8	40.9646	B8	2.5603	78	0.1600	38	0.0100
F7	39.2578	B7	2.4536	77	0.1534	37	0.0096
F6	37.5509	B6	2.3469	76	0.1467	36	0.0092
F5	35.8441	B5	2.2403	75	0.1400	35	0.0088
F4	34.1372	B4	2.1336	74	0.1333	34	0.0083
F3	32.4303	B3	2.0269	73	0.1267	33	0.0079
F2	30.7235	B2	1.9202	72	0.1200	32	0.0075
F1	29.0166	B1	1.8135	71	0.1133	31	0.0071
F0	27.3098	B0	1.7069	70	0.1067	30	0.0067
EF	26.4563	AF	1.6535	6F	0.1033	2F	0.0065
EE	25.6029	AE	1.6002	6E	0.1000	2E	0.0063
ED	24.7495	AD	1.5468	6D	0.0967	2D	0.0060
EC	23.8960	AC	1.4935	6C	0.0933	2C	0.0058
EB	23.0426	AB	1.4402	6B	0.0900	2B	0.0056
EA	22.1892	AA	1.3868	6A	0.0867	2A	0.0054
E9	21.3358	A9	1.3335	69	0.0833	29	0.0052
E8	20.4823	A8	1.2801	68	0.0800	28	0.0050
E7	19.6289	A7	1.2268	67	0.0767	27	0.0048
E6	18.7755	A6	1.1735	66	0.0733	26	0.0046
E5	17.9220	A5	1.1201	65	0.0700	25	0.0044
E4	17.0686	A4	1.0668	64	0.0667	24	0.0042
E3	16.2152	A3	1.0134	63	0.0633	23	0.0040
E2	15.3617	A2	0.9601	62	0.0600	22	0.0038
E1	14.5083	A1	0.9068	61	0.0567	21	0.0035
E0	13.6549	A0	0.8534	60	0.0533	20	0.0033
DF	13.2282	9F	0.8268	5F	0.0517	1F	0.0032
DE	12.8015	9E	0.8001	5E	0.0500	1E	0.0031
DD	12.3747	9D	0.7734	5D	0.0483	1D	0.0030
DC	11.9480	9C	0.7468	5C	0.0467	1C	0.0029
DB	11.5213	9B	0.7201	5B	0.0450	1B	0.0028
DA	11.0946	9A	0.6934	5A	0.0433	1A	0.0027
D9	10.6679	99	0.6667	59	0.0417	19	0.0026
D8	10.2412	98	0.6401	58	0.0400	18	0.0025
D7	9.8144	97	0.6134	57	0.0383	17	0.0024
D6	9.3877	96	0.5867	56	0.0367	16	0.0023
D5	8.9610	95	0.5601	55	0.0350	15	0.0022
D4	8.5343	94	0.5334	54	0.0333	14	0.0021
D3	8.1076	93	0.5067	53	0.0317	13	0.0020
D2	7.6809	92	0.4801	52	0.0300	12	0.0019
D1	7.2542	91	0.4534	51	0.0283	11	0.0018
D0	6.8274	90	0.4267	50	0.0267	10	0.0017
CF	6.6141	8F	0.4134	4F	0.0258	0F	0.0016
CE	6.4007	8E	0.4000	4E	0.0250	0E	0.0016
CD	6.1874	8D	0.3867	4D	0.0242	0D	0.0015
CC	5.9740	8C	0.3734	4C	0.0233	0C	0.0015
CB	5.7607	8B	0.3600	4B	0.0225	0B	0.0014
CA	5.5473	8A	0.3467	4A	0.0217	0A	0.0014
C9	5.3339	89	0.3334	49	0.0208	09	0.0013
C8	5.1206	88	0.3200	48	0.0200	08	0.0013
C7	4.9072	87	0.3067	47	0.0192	07	0.0012
C6	4.6939	86	0.2934	46	0.0183	06	0.0011
C5	4.4805	85	0.2800	45	0.0175	05	0.0011
C4	4.2672	84	0.2667	44	0.0167	04	0.0010
C3	4.0538	83	0.2534	43	0.0158	03	0.0010
C2	3.8404	82	0.2400	42	0.0150	02	0.0009
C1	3.6271	81	0.2267	41	0.0142	01	0.0009
C0	3.4137	80	0.2134	40	0.0133	00	0.0008

表8-5 設定できるLFOの発振周波数

● PMS/AMS: PHASE MODULATION SENSITIVITY/AMPLITUDE MODULATION SENSITIVITY (38H~3FH)

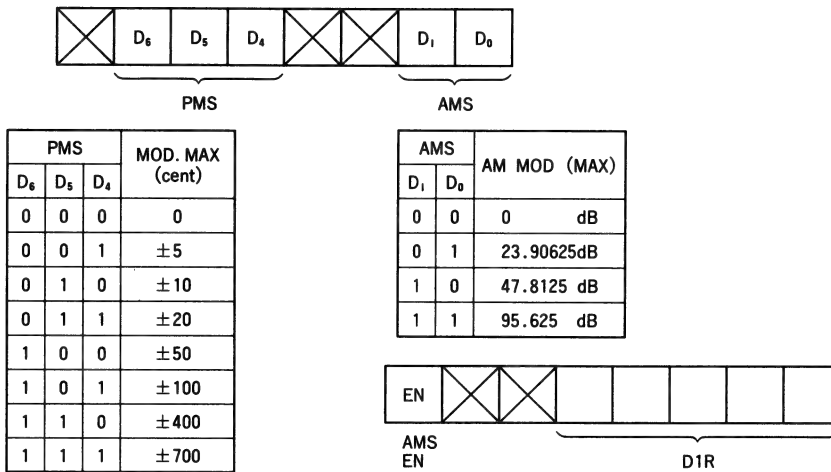


図8-13

PMSはLFOの出力を8ビットコードをKC, KFに加えることによって、ビブラート、ゆらぎのある音を作ります。感度は0を含めて8段階設定できます。AMSはLFAデータに従ってエンベロープジェネレータが振幅変調を行うときの最大変調度を設定します。この振幅変調は、AMS-ENスイッチにより、スロットごとに変調をかけるかどうかを選択できます。このAMSのデータはチャンネルごとに設定します。

● PMD/AMD: PHASE MODULATION DEPTH/AMPLITUDE MODULATION DEPTH (19H)

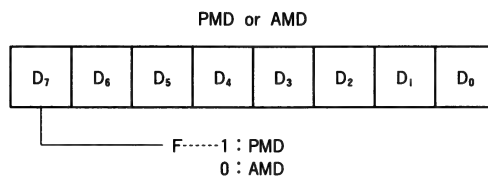


図8-14

AM, PMの変調度を微調整することができます。負の値にすると、波形の極性が逆になります。ビット7で周波数変調と振幅変調の切り換えを行います。残りのビットで出力レベルを1/128の分解能で設定します。なお、PMDの時は2の補数、AMDはバイナリになります。

(3) PG: PHASE GENERATOR

FM音源では、PSGと異なり基準クロックを最大分周した時の周波数を基準にして、出力したい周波数がこの何倍であるかという倍率をあたえて出力したい周波数を設定します。こうすることにより、PSGに比べ高い周波数での誤差が小さくなります。また、キャリア周波数、モジュレータ周波数を設定し、ビブラート効果、ノイズの発生を行うことができます。

● KC : KEY CORD (28H~2FH)

X	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
	OCT			NOTE			

OCT			オクターブ
D ₆	D ₅	D ₄	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

NOTE				NOTE
D ₃	D ₂	D ₁	D ₀	
0	0	0	0	C #
0	0	0	1	D
0	0	1	0	D #
0	1	0	0	E
0	1	0	1	F
0	1	1	0	F #
1	0	0	0	G
1	0	0	1	G #
1	0	1	0	A
1	1	0	0	A #
1	1	0	1	B
1	1	1	0	C

図8-15

7ビット中上位3ビットがオクターブを表し、下位4ビットがNOTEを表します。

● KF : KEY FRACTION (30H~37H)

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	X	X
KEY FRACTION							

KEY FRACTION						KF (Cent)
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	
0	0	0	0	0	0	0
0	0	0	0	0	0	
0	0	0	0	0	0	
0	1	0	0	0	0	25
0	1	0	0	0	1	
0	1	0	0	0	0	
0	1	0	0	0	0	
1	0	0	0	0	0	50
1	0	0	0	0	0	
1	0	0	0	0	0	
1	1	0	0	0	0	75
1	1	0	0	0	1	
1	1	0	0	0	0	
1	1	1	1	1	1	
0	0	0	0	0	0	00
0	0	0	0	0	0	

図8-16

1度の音程差(100セント)を約1.6セントステップで位相情報を設定します。

● MUL : PHASE MULTIPLY (40H~5FH)

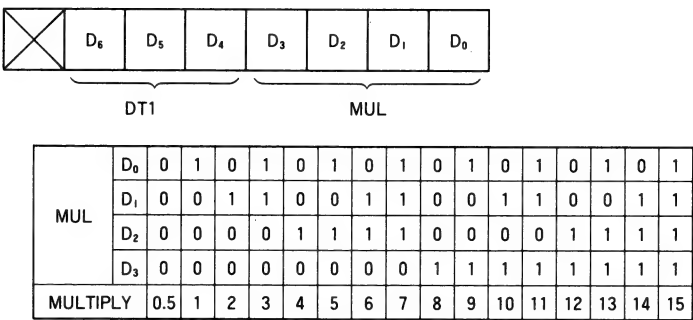


図8-17

KC, KF の位相情報に倍率をかけた位相情報を設定します。

● DT1 : DETUNE (1) (40H~5FH)

ディチューンとは、各オペレータの周波数にずれをあたえて音に厚みをつけることです。ディチューン1では、KC, KF で設定した位相情報と少し周波数のずれた位相情報を設定します。キーコードによりスケーリングされます。

● DT2 : DETUNE (2) (C0H~DFH)

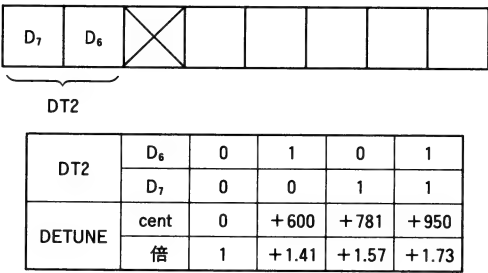


図8-18 ディチューン 2

ディチューン2では、KC, KF で設定した位相情報と大きく周波数のずれた位相情報を設定します。これは効果音の発生に有効です。キーコードによりスケーリングされます。

OCT					DT1							
					D-CENT				D-FREQ (Hz)			
D ₆	D ₅	D ₄	D ₃	D ₂	0	1	2	3	0	1	2	3
0	0	0	0	0	0.000	0.000	5.025	10.036	0.000	0.000	0.053	0.107
0	0	0	0	1	0.000	0.000	4.228	8.445	0.000	0.000	0.053	0.107
0	0	0	1	0	0.000	0.000	3.559	7.110	0.000	0.000	0.053	0.107
0	0	0	1	1	0.000	0.000	2.993	5.980	0.000	0.000	0.053	0.107
0	0	1	0	0	0.000	2.515	5.025	5.025	0.000	0.053	0.107	0.107
0	0	1	0	1	0.000	2.115	4.228	6.338	0.000	0.053	0.107	0.160
0	0	1	1	0	0.000	1.778	3.555	5.330	0.000	0.053	0.107	0.160
0	0	1	1	1	0.000	1.496	2.990	4.483	0.000	0.053	0.107	0.160
0	1	0	0	0	0.000	1.258	2.515	5.025	0.000	0.053	0.107	0.213
0	1	0	0	1	0.000	1.057	3.170	4.225	0.000	0.053	0.160	0.213
0	1	0	1	0	0.000	0.889	2.667	3.555	0.000	0.053	0.160	0.213
0	1	0	1	1	0.000	0.748	2.242	3.735	0.000	0.053	0.160	0.267
0	1	1	0	0	0.000	1.258	2.515	3.143	0.000	0.107	0.213	0.267
0	1	1	0	1	0.000	1.057	2.114	3.170	0.000	0.107	0.213	0.320
0	1	1	1	0	0.000	0.889	1.778	2.667	0.000	0.107	0.213	0.320
0	1	1	1	1	0.000	0.748	1.869	2.615	0.000	0.107	0.267	0.373
1	0	0	0	0	0.000	0.629	1.572	2.515	0.000	0.107	0.267	0.427
1	0	0	0	1	0.000	0.793	1.586	2.114	0.000	0.160	0.320	0.427
1	0	0	1	0	0.000	0.667	1.334	2.001	0.000	0.160	0.320	0.480
1	0	0	1	1	0.000	0.561	1.308	1.869	0.000	0.160	0.373	0.533
1	0	1	0	0	0.000	0.629	1.258	1.729	0.000	0.213	0.427	0.587
1	0	1	0	1	0.000	0.529	1.057	1.586	0.000	0.213	0.427	0.640
1	0	1	1	0	0.000	0.445	1.001	1.445	0.000	0.213	0.480	0.693
1	0	1	1	1	0.000	0.467	0.935	1.308	0.000	0.267	0.533	0.747
1	1	0	0	0	0.000	0.393	0.865	1.258	0.000	0.267	0.587	0.853
1	1	0	0	1	0.000	0.397	0.793	1.123	0.000	0.320	0.640	0.907
1	1	0	1	0	0.000	0.334	0.723	1.056	0.000	0.320	0.693	1.013
1	1	0	1	1	0.000	0.327	0.654	0.935	0.000	0.373	0.747	1.067
1	1	1	0	0	0.000	0.315	0.629	0.865	0.000	0.427	0.853	1.173
1	1	1	0	1	0.000	0.264	0.562	0.865	0.000	0.427	0.907	1.173
1	1	1	1	0	0.000	0.250	0.528	0.865	0.000	0.480	1.013	1.173
1	1	1	1	1	0.000	0.234	0.467	0.865	0.000	0.533	1.067	1.173

表8-6 ディチューン1

(4) OP:FM OPERATOR

フェイズジェネレータからの位相情報にしたがってサインテーブルを読みだします。

● CON:CONNECTION (20H~27H)

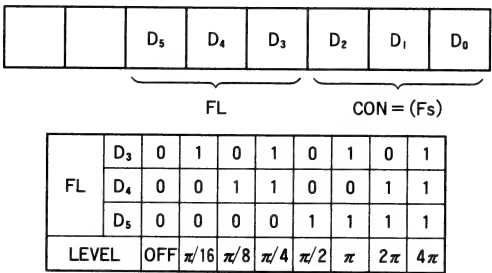


図8-19

アルゴリズムを設定して8通り FM-OP 回路構成にでき、それぞれ異なる音色にできます。アルゴリズムは、8-2-1の図を見て下さい。

● FL:SELF FEEDBACK LEVEL (20H~27H)

フィードバックとは、オペレータの出力を再び同じオペレータの入力とすることです。硬い音を出すときは強く、柔らかい音を出すときは弱くかけます。また、フィードバックをかけるだけで強い変調がかかるので、過変調を利用してホワイトノイズなどの効果音に用いることができます。なお、フィードバックのレベルは各音ごとに設定できます。

(5) EG:ENVELOPE GENERATOR

オペレータが“正弦波読み出し部”の信号にエンベロープジェネレータの出力をかけ合わせ音色、音量の時間的変化を作りだします。この変化はアタック部では指数式的、ディケイ部ではすべて直線的变化です。

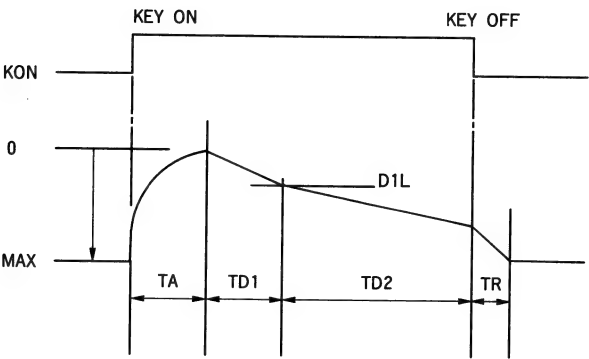


図8-20 エンベロープ

● KON : KEY ON (08H)

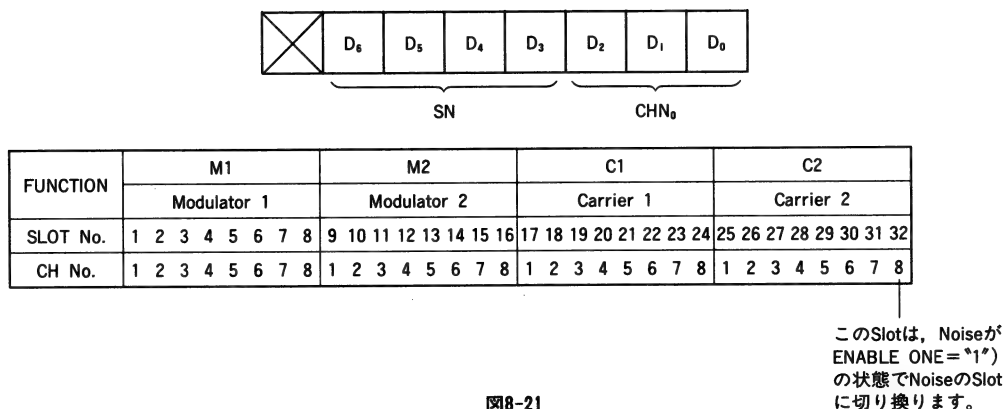


図8-21

チャンネルとスロットを指定し、キーオン(1)することにより音源ボードがアクセスされ、エンベロープジェネレータが動作し始めます。キーオンしたあとは適当な時間の後にキーオフ(0)します。スロットナンバーのD3, D4, D5, D6ビットは同じチャンネルのスロットナンバーの番号の小さい方から順に対応しています。

● AR : ATTACK RATE (80H~9FH)

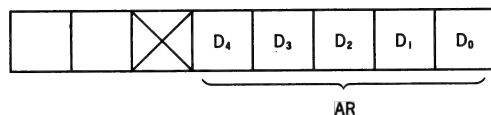


図8-22

キーオンから音量が最大になるまでの速さを設定します。設定値が大きいくほど速く音量が最大になります。また、キーコードによりスケーリングが可能です。スケーリングとは、出力したい楽器音の音量やエンベロープが音域によって大きく変化する場合に行う補正のことです。

● D1R : FIRST DECAY RATE (A0H~BFH)

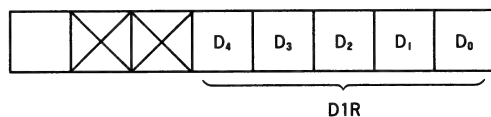


図8-23

音量が最大からD1Lに下がる速さを設定します。D1Rはキーコードによりスケーリングされます。D1Rはキーコードによりスケーリングできます。

● D2R : SECOND DECAY RATE (C0H ~ DFH)

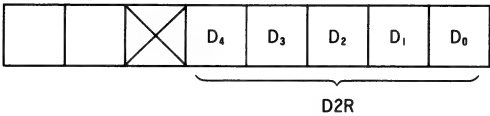


図8-24

キーオン中の状態で音量がD1Lから下がるまでの速さを設定します。D2Rはキーコードによりスケーリングされます。

● RR : RELEASE RATE (E0H ~ FFH)

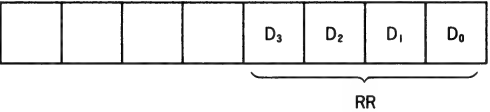


図8-25

キーオフした直後の状態からの音量の下がり速さを設定します。すなわち、余韻の状態を設定します。RRはキーコードによりスケーリングされます。

● D1L : FIRST DECAY LEVEL (E0H ~ FFH)

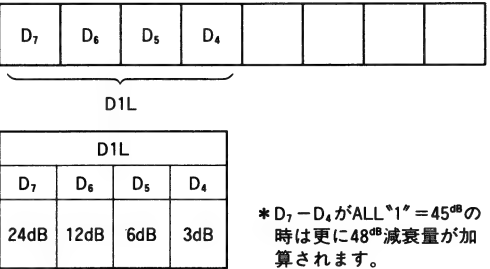


図8-26

EGはこのレベルを経過するとファーストディケイからセカンドディケイに移ります。分解能は3dBです。

● TL : TOTAL LEVEL (60H ~ 7FH)

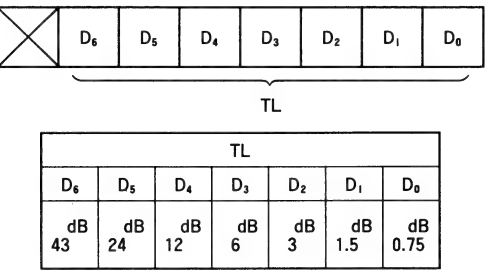


図8-27 TOTAL LEVELの各ビットと重点付け

EG で演奏された各時点での値についてトータルレベルを加算してオペレータに出力し音色(変調度)と音量をコントロールします。最小分解能は 0.75dB です。

● KS : KEY SCALING (80H~9FH)

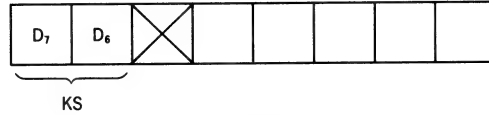


図8-28

AR, D1R, D2R, RR のスケーリングレベルを設定します。

- (*) キースケーリングされた後のRATEは下式の様に入力レート (R) の 2 倍に下表の値 (R_{KS}) を加えたものであります。
- (**) AR, D1R, D2Rはレジスタに書き込んだ値を入力レート (R) としますがRRはレジスタに書き込んだ値の 2 倍に 1 を加えた値を入力レート (R) として計算します。

$$\text{RATE} = 2 * R + R_{KS}$$

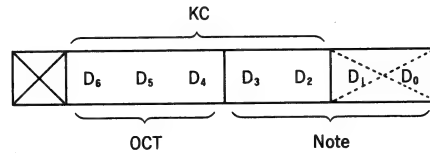
計算結果が63より大きな値の時は全て
RATE=63とします。

◇R: 入力の各レート (**)

◇R_{KS}: KEY CODEとKSで定まる下表による値

◇但しここでのKEY CODEは下図の様にNoteの下位 2 ビットは切り捨てたKCを用います。

KC	KS 0	KS 1	KS 2	KS 3
0	0	0	0	0
1	0	0	0	1
2	0	0	1	2
3	0	0	1	3
4	0	1	2	4
5	0	1	2	5
6	0	1	3	6
7	0	1	3	7
8	1	2	4	8
9	1	2	4	9
10	1	2	5	10
11	1	2	5	11
12	1	3	6	12
13	1	3	6	13
14	1	3	7	14
15	1	3	7	15
16	2	4	8	16
17	2	4	8	17
18	2	4	9	18
19	2	4	9	19
20	2	5	10	20
21	2	5	10	21
22	2	5	11	22
23	2	5	11	23
24	3	6	12	24
25	3	6	12	25
26	3	6	13	26
27	3	6	13	27
28	3	7	14	28
29	3	7	14	29
30	3	7	15	30
31	3	7	15	31



スケーリングした後の各レートでアタック、ファストディケイ、セカンドディケイ、リリースの各時間が定まります。

図8-29

第8章 サウンド機能

* 図8-29 で決まるキースケーリング後のRATE 6ビットを、上位4ビット、下位2ビットに分けて表現してあります。

* (10%－90%) or (90%－10%)の表は、レベルが10%から90%又は、90%から10%に達する時間を示します。

* (96dB－0dB) or (0dB－96dB)の表は、レベルが0%から100%又は、100%から0%に達する時間を示します。

*注) この表は、 $\phi_M = 3.6M_{Hz}$ で計算してあります。

... EG ATTACK TIME EG DECAY TIME EG ATTACK TIME EG DECAY TIME ...	
RATE	mSEC(10%－90%)	RATE	mSEC(90%－10%)	RATE	mSEC(96dB－0dB)	RATE	mSEC(0dB－96dB)
15 3	0.00	15 3	1.36	15 3	0.00	15 3	6.73
15 2	0.27	15 2	1.36	15 2	0.53	15 2	6.73
15 1	0.27	15 1	1.36	15 1	0.53	15 1	6.73
15 0	0.27	15 0	1.36	15 0	0.53	15 0	6.73
14 3	0.34	14 3	1.55	14 3	0.64	14 3	7.69
14 2	0.39	14 2	1.81	14 2	0.75	14 2	8.97
14 1	0.47	14 1	2.18	14 1	0.90	14 1	10.76
14 0	0.59	14 0	2.72	14 0	1.12	14 0	13.45
13 3	0.62	13 3	3.11	13 3	1.22	13 3	15.38
13 2	0.73	13 2	3.63	13 2	1.42	13 2	17.94
13 1	0.87	13 1	4.35	13 1	1.71	13 1	21.53
13 0	1.09	13 0	5.44	13 0	2.13	13 0	26.91
12 3	1.25	12 3	6.22	12 3	2.22	12 3	30.75
12 2	1.46	12 2	7.25	12 2	2.60	12 2	35.88
12 1	1.75	12 1	8.70	12 1	3.11	12 1	43.05
12 0	2.19	12 0	10.88	12 0	3.89	12 0	53.81
11 3	2.50	11 3	12.43	11 3	4.45	11 3	61.50
11 2	2.92	11 2	14.51	11 2	5.19	11 2	71.75
11 1	3.50	11 1	17.41	11 1	6.23	11 1	86.10
11 0	4.37	11 0	21.76	11 0	7.79	11 0	107.63
10 3	5.00	10 3	24.87	10 3	8.90	10 3	125.00
10 2	5.83	10 2	29.01	10 2	10.38	10 2	143.50
10 1	7.00	10 1	34.82	10 1	12.46	10 1	172.20
10 0	8.75	10 0	43.52	10 0	15.57	10 0	215.25
9 3	10.00	9 3	49.74	9 3	17.80	9 3	246.00
9 2	11.66	9 2	55.03	9 2	20.76	9 2	287.00
9 1	13.99	9 1	69.63	9 1	24.92	9 1	344.41
9 0	17.49	9 0	87.04	9 0	31.15	9 0	430.51
8 3	19.99	8 3	99.47	8 3	35.60	8 3	492.01
8 2	23.22	8 2	116.05	8 2	41.53	8 2	574.01
8 1	27.39	8 1	139.26	8 1	49.83	8 1	698.81
8 0	34.97	8 0	174.08	8 0	62.29	8 0	861.01
7 3	39.98	7 3	198.95	7 3	71.19	7 3	984.02
7 2	46.65	7 2	232.11	7 2	83.06	7 2	1148.02
7 1	55.98	7 1	278.53	7 1	99.67	7 1	1377.62
7 0	69.97	7 0	348.16	7 0	124.59	7 0	1722.03
6 3	79.97	6 3	397.90	6 3	142.38	6 3	1958.03
6 2	93.30	6 2	464.21	6 2	166.12	6 2	2296.04
6 1	111.96	6 1	557.06	6 1	199.34	6 1	2755.24
6 0	139.95	6 0	696.32	6 0	249.17	6 0	3444.05
5 3	159.94	5 3	795.79	5 3	284.77	5 3	3936.06
5 2	186.60	5 2	928.43	5 2	332.23	5 2	4592.07
5 1	223.91	5 1	1114.11	5 1	398.68	5 1	5510.49
5 0	279.89	5 0	1392.64	5 0	498.35	5 0	6538.11
4 3	319.88	4 3	1591.59	4 3	569.54	4 3	7872.12
4 2	373.19	4 2	1856.85	4 2	664.46	4 2	9154.14
4 1	447.83	4 1	2228.22	4 1	797.35	4 1	11020.97
4 0	559.79	4 0	2785.28	4 0	996.69	4 0	13776.21
3 3	639.76	3 3	3183.18	3 3	1139.08	3 3	15744.24
3 2	746.38	3 2	3713.71	3 2	1328.92	3 2	18338.28
3 1	895.66	3 1	4456.45	3 1	1594.71	3 1	22041.94
3 0	1119.57	3 0	5570.56	3 0	1993.39	3 0	27552.43
2 3	1279.51	2 3	6366.35	2 3	2278.16	2 3	31499.49
2 2	1492.76	2 2	7427.41	2 2	2657.85	2 2	36736.57
2 1	1791.32	2 1	8912.90	2 1	3189.42	2 1	44093.88
2 0	2239.15	2 0	11141.12	2 0	3986.77	2 0	55104.85
1 3	2559.02	1 3	12732.71	1 3	4556.31	1 3	62976.98
1 2	2985.53	1 2	14854.83	1 2	5315.70	1 2	73473.14
1 1	3582.63	1 1	17825.79	1 1	6378.84	1 1	88167.77
1 0	4478.29	1 0	22282.24	1 0	7973.55	1 0	110209.71
0 3	INFINITY	0 3	INFINITY	0 3	INFINITY	0 3	INFINITY
0 2	INFINITY	0 2	INFINITY	0 2	INFINITY	0 2	INFINITY
0 1	INFINITY	0 1	INFINITY	0 1	INFINITY	0 1	INFINITY
0 0	INFINITY	0 0	INFINITY	0 0	INFINITY	0 0	INFINITY

表8-7

(6) ノイズ

ノイズの質はノイズジェネレータのクロックを外部からのコントロールにより変化させることができます。

● NE : NOISE ENABLE (0FH)

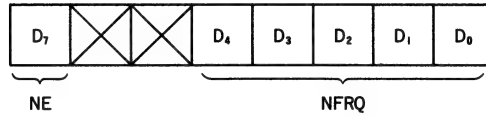


図8-30

ビット7を1とすれば32番目のスロットはノイズとなります。

● NFRQ : NOISE FREQUENCY (0FH)

ノイズ周波数の設定します。

NFRQとノイズ周波数の関係は

$$f_{\text{Noise}} = \frac{f_M}{32 * (NFRQ)} \quad \diamond f_M = 3579.545 \text{ KHZ} \quad (\text{YM2151に}\text{加える}\text{クロック}\text{周波数})$$

となり約3.5KHzから約111.9KHzの間変化させることができます。このときノイズの周期は

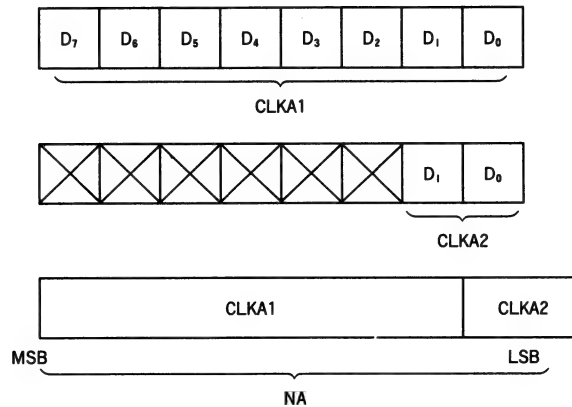
$$T_{\text{Noise}} = \frac{217-1}{f_{\text{Noise}}(\text{HZ})}$$

により求めることが出来、約37.5SECから約1.17SECの値になります。

(7) タイマー

このFM音源には10ビットのプリセットブルタイマーA, 8ビットのプリセットブルタイマーBがありオーバーフロー時にデータバスにフラグを立て割り込みを行うことができます。

● CLKA (10H, 11H)



$$T_A^{(ms)} = \frac{64 * (1024 - NA)}{f_M(\text{KHZ})} \quad * f_M = 3579.545 \text{ KHZ} \quad (\text{YM2151に}\text{加える}\text{クロック}\text{周波数})$$

図8-31

10ビットのプリセッタブルタイマでオーバーフローをしたときはCPUにIRQ割り込みをかけることができます。デバイス内部でオーバーフローしたとき、キーオン信号として利用できます。

● CLKB(12H)

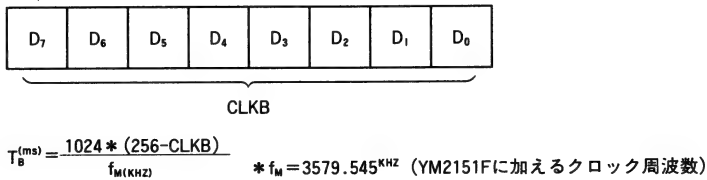


図8-32

8ビットのプリセッタブルタイマでオーバーフローをしたときはCPUにIRQ割り込みをかけることができます。

● IRQEN, LOAD, F RESET, CSM(14H)

IRQEN：タイマーが発生するオーバーフローをフラグレジスタにレジストすることを可能にし割り込み要求も可能にします。

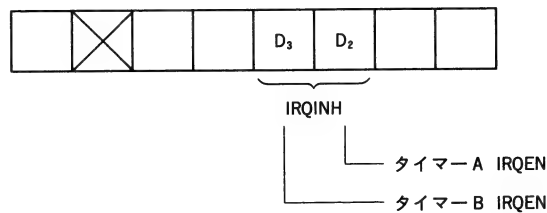


図8-33

LOAD：各タイマの始動、停止をコントロールします。

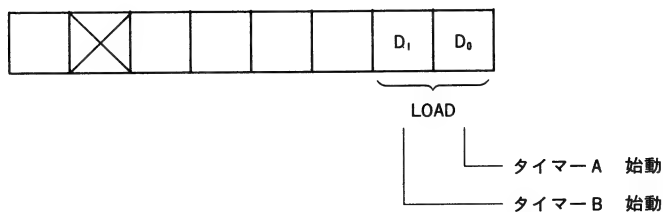


図8-34

F RESET：各タイマがオーバーフローしたことを示すフラグレジスタをリセットします。オーバーフロー時のIRQ割り込みをするか、しないかを決めます。

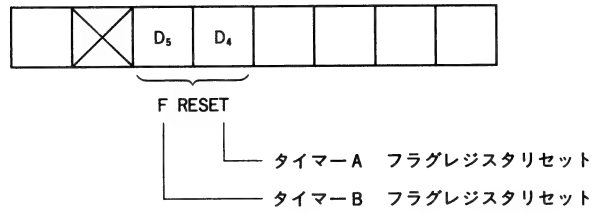


図8-35

CSM：オーバーフロー時に音源のすべてのスロットをキーオンできます。

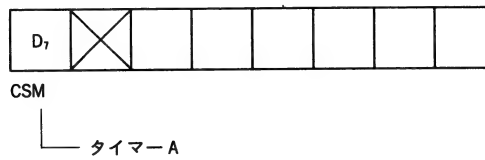
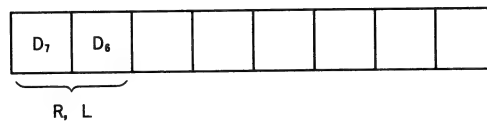


図8-36

(8) ACC: ACCUMULATOR

RL レジスタによりオペレータからのデータをL系列, R系列に入力して加算します。加算されたデータは所定のフォーマットでシリアルに出力されます。

● RL: RIGHT/LEFT (20H~27H)



D ₇	D ₆	サウンド出力
1	0	Right
0	1	Left
1	1	R/L同時表示
0	0	出力無し

図8-37

R/Lの、出力チャンネルを設定します。

各8つのチャンネルのR/Lオーディオ出力の設定はビット6, 7によって行います。

(B) リードモード

YM2151には、読み出しレジスタが1つあります。レジスタ番号を指定する必要はありません。

● B: WRITE BUSY FLAG

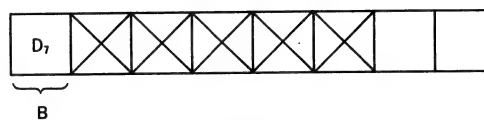


図8-38

書き込み中は1となるフラグです。データを続けて書き込むときは、このフラグが0になったことを確認しながら行います。

● IST

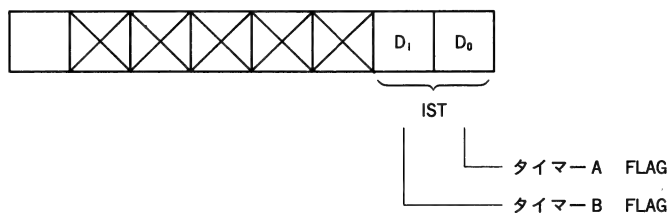


図8-39

フラグレジスタのフラグ状態を示します。タイマのオーバーフローにより IST のいずれかのビットが1になると IRQ 端子がLになります。

8-2-4 キーコードの補正

YM2151(OPM)は、約3.58MHzのシステムクロックで使用したときに、KEY FRACTION (KF)値が0となり、最もずれの少ない音程となるようにデバイス内部にデータを持っています。しかし、X1ではシステムクロックが4MHzのため音程が約192セント高くなるため補正を必要とします。

補正法：KF=5 または 6 を与えて、ずれを約200セントにします。このとき NOTE データと発音音程の関係は

NOTE	0	1	2	4	5	6	8	9	10	12	13	14
音程	D#	E	F	F#	G	G#	A	A#	B	C#	C	D

C	D	E	F	G	A	B	C
ド	レ	ミ	ファ	ソ	ラ	シ	ド

図8-40

のように1音高い音になります。

8-2-5 テンポの設定

テンポ設定は CTC による割り込み処理プログラムで行うことができます。X1turbo シリーズでは内蔵の CTC, X1 シリーズでは CZ-8BS1 の CTC を使用します。YM2151 内のタイマは使用できません。

内蔵 CTC:I/O ポート(1FA0H)

CZ-8BS1 の CTC:I/O ポート(704H)

テンポ設定を行うには、CTC のチャンネル 0 はタイマモードに、チャンネル 3 はカウンタモードにします。このとき割り込み周期は

$$T = TC0 \times TC \times P \times TC3$$

TC0 : チャンネル 0 時間定数

TC : システムクロック周期, 250nsec

P : プリスケアラ値

TC3 : チャンネル 3 時間定数

となります。

```

DI
LD   BC, CTC
LD   A, 00010101B   ;* 1.
OUT  (C), A
LD   A, 0FAH        ;時定数
OUT  (C), A
LD   BC, CTC+3
LD   A, VECTER      ;下位ベクターアドレス
OUT  (C), A
LD   A, 11010101B   ;* 2.
OUT  (C), A
LD   A, TIME        ;時定数設定
OUT  (C), A
EI

```

* 1.	ビット 7	割り込み禁止
	ビット 6	タイムモード選択
	ビット 5	プリスケアラ値 16
	ビット 4	don't care
	ビット 3	時間定数をロード後タイマーは動作する
	ビット 2	次に書き込まれるデータは時間定数データ
	ビット 1	チャンネルは現在の動作を継続する

* 2.	ビット 6	カウンタモード選択
	ビット 5	カウンタモード時 don't care
	ビット 4	CLK/TRGの立ち上がりでダウンカウンタを-1 (本ボードではチャンネル0の時間定数が0の時)

表8-8 テンポの設定例

第 9 章

各種インターフェイス

X1turbo には、プリンターインターフェイス、RS-232C インターフェイスなどが標準装備されています。(X1 では一部オプション)この章には、各インターフェイスの、主としてハードウェア的な側面がまとめてあります。

9-1 プリンターインターフェイス

X1 シリーズでは 8 ビットパラレル転送用出力端子がプリンターインターフェイス用に標準装備されています。

9-1-1 セントロニクスインターフェイス

セントロニクスインターフェイスは、米国のセントロニクス社が、自社のプリンター用に考案した規格です。X1 シリーズのプリンターインターフェイスはこのセントロニクス規格に、ほぼ準拠しています。といっても省略したピンがあり、タイミングも若干違うので、純正品以外の「セントロニクス規格準拠」プリンターでは、直接接続できない場合があります。

ピンNo.	信 号 名	方 向※	機 能
1	STROBE	出 力	データ・ストロブ・パルス
2	DATA 0	出 力	8 ビット・パラレル・データ
3	" 1		
4	" 2		
5	" 3		
6	" 4		
7	" 5		
8	" 6		
9	" 7		
10	ACKNLG	入 力	データ認知パルス
11	BUSY	入 力	プリンター動作中ステータス
12	PE	入 力	用紙有無ステータス
13	SLCT	入 力	プリンター・セレクト・ステータス
14	N. C		ノン・コネクション
15	N. C		
16	0V	————	ロジック・GND
17	FG	————	フレーム・GND
18	N. C		

19	PAIR GND	————	1～12ピンのツイスト・ペア線のGND
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31	INIT	出 力	プリンター・イニシャライズ
32	FAULT	入 力	オフライン or 用紙切れ
33	N. C		
34	N. C		
35	FUSE	入 力	ヒューズ断
36	N. C		

※方向はコンピュータ本体から見たとき

(a) セントロニクスインターフェイス

端子番号	信 号 名		端子番号	信 号 名	
1	STROBE	ストロブ信号	8	PA 6	パラレルデータ
2	PA 0	パラレルデータ	9	PA 7	"
3	PA 1	"	10	N. C.	非接続
4	PA 2	"	11	BUSY	ビジー信号
5	PA 3	"	12	N. C.	非接続
6	PA 4	"	13	GND	グラウンド
7	PA 5	"	14	GND	グラウンド

(b) X1プリンターインターフェイス

表9-1 セントロニクスインターフェイスとX1プリンターインターフェイス

9-1-2 8255 ②のビット構成

プリンターインターフェイスには8255(PPI: Programable Peripheral Interface)が使用されています。この8255はA, B, Cの3つの8ビットのポートを持ち、そのうちA, Bの2つのポートをそれぞれ次の3つのモードで使用できます。

- MODE 0 : 制御信号なしの入出力ポート
- MODE 1 : 制御信号つきの入出力ポート
- MODE 2 : 制御信号つきの双方向性の入出力ポート (ポートAのみ)

プリンターインターフェイスでは、MODE1 を使用しています。8255②のポートの中で、プリンターインターフェイスに直接に関係するのは次の3つです。

ポートA(I/Oアドレス1A00H):プリンターに出力される8ビットデータ

ポートB(I/Oアドレス1A01H):BUSY(ビット3)

ポートC(I/Oアドレス1A02H):STROBE(ビット7)

9-1-3 制御信号

X1シリーズではBUSY, STROBE という2本の制御線によってプリンターにデータを送信しています。

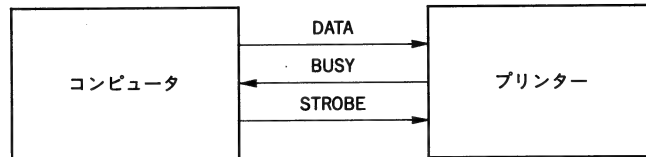


図9-1 プリンターとコンピュータの接続図

(1) BUSY(プリンター→コンピュータ)

BUSY 信号(プリンター動作中ステータス)は、プリンターが内部処理中のためデータを受信できない状態あること示す信号です。BUSY 信号がHのときは、Lに変わるまで待機します。

(2) STROBE(コンピュータ→プリンター)

STROBE 信号(データストロブパルス)は、コンピュータからプリンターへの信号で、プリンターへの出力データの値が確定し、データの取り込みが可能であることを知らせるLのワンショットパルスです。プリンターは、この信号が来たらデータを読み込みます。

9-1-4 プリンターインターフェイスの動作

プリンターに、データを送出する手順は次のようになります。

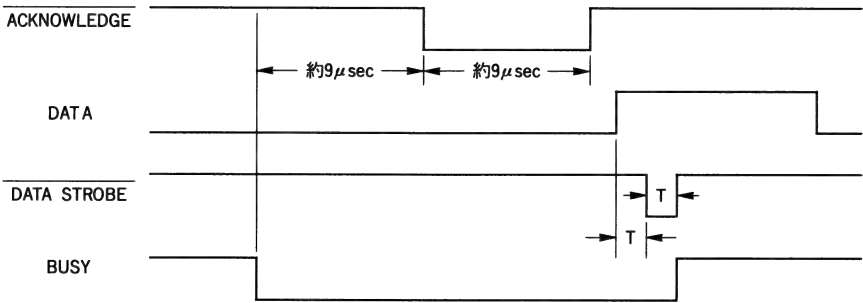
- (1) BUSY 信号がLであるかを調べ、もしHであるならばLになるまで待機する。
- (2) ポートAに、プリンターに送信するデータをセットする。
- (3) STROBE 信号をLにする。
- (4) 一定時間後(1~30 μ sec)に STROBE 信号をHにする。
- (5) (1)に戻る。

CZ-8PD2(シャープ専用ドットプリンター)ではコンピュータ本体からのストロブ信号のLからHへの立ち上がりエッジでデータをラッチします。

本来セントロニクス規格では、STROBE 信号と、BUSY 信号、及び ACKLNG 信号の3つの信号でハンドシェイクを行ないます。(ハンドシェイクとは、送信側と受信側とで互いに信号を出し合ってデータの転送を確認しながら行う方法です。)X1シリーズのプリンターインターフェイスでは、ACKLNG 信号が省略されているため、BUSY 信号のみを頼りにタイミングを取らなければなりません。プリンターによっては、STROBE 信号をHに戻してから、しばらく待たないとBUSY 信号がHにならないことがあるので注意して下さい。このようなプリンターでは、BUSY 信号の読み込みが速すぎるとデータが欠けることがあります。これを避けるには、上記の手順の(4)のあと、適当な長さの待ち時間を挿入します。

ポートCにデータストローブパルスを出すときは、ポートCに直接値を書き込まないで、ポートCビットセット／リセットコマンドをコントロールレジスタ(I/O アドレス 1A03H)に書き込むと、ビット7の状態のみを変えることができます。

- ① データ転送速度.....1000～6000CPS
- ② 同期方式.....外部供給ストローブパルスによる
- ③ ハンドシェイク.....ACKNOWLEDGE, BUSY信号による
- ④ ロジックレベル.....TTLレベルにコンパチブル



T : 0.5μsec以上

図9-2 CZ-8PDデータ受信タイミング

9-2 ジョイスティックインターフェイス

ジョイスティックは、スティック(レバー)を倒した方向をX方向、Y方向にそれぞれ設けられたスイッチによって読み取ります。

X1 シリーズにはアタリ社規格のジョイスティック用端子が2つ標準装備されています。これらの端子は、それぞれ PSG (AY-3-8910) の2つの8ビット汎用 I/O ポートに接続されています。

ジョイスティックインターフェイスを操作するときは、PSG の内部レジスタ R7, R14, R15 を使用します。

レジスタ	機 能	7	6	5	4	3	2	1	0	備 考
7	各チャンネル音量ON/OFF	IN/OUT		ノイズ			トーン			1 ON 0 OFF
	ジョイスティックポート入出力スイッチ	B	A	C	B	A	C	B	A	1 ON 0 OUT
14	ジョイスティック I/O ポート A	00H～FFH								8ビット パラレル データ
15	ジョイスティック I/O ポート B	00H～FFH								

表9-2 PSGの各レジスタのビット構成

ポートの入出力は R7 で設定します。
R 7 : ビット 6 (A ポート) L-入力 H-出力
 ビット 7 (B ポート) L-入力 H-出力

I/Oポート* ビット番号	ジョイスティック 端子番号 (機能)
0	1 (前)
1	2 (後)
2	3 (左)
3	4 (右)
4	5 (—)
5	6 (トリガ1)
6	7 (トリガ2)
7	9 (—)
(GND)	8 (GND)

*) ジョイスティック1がポートA
ジョイスティック2がポートB

表9-3 ジョイスティックインターフェース

ジョイスティックで状態センスを行う場合は必ずI/Oポートを入力に設定して下さい。出力として使用しますと、PSGを破壊するおそれがあります。

9-3 マウスインターフェイス

グラフィックツールなどの座標入力等においては、マウスが大いに威力を発揮します。マウスは、底のボールの回転により、X-Y方向の移動方向を読み取ります。この信号をカウンタに加えることによって、X-Y方向の移動量を知るようになっています。

9-3-1 制御信号

X1シリーズでは、データをRS-232Cと同様の信号形式で送出するタイプのマウスが使われています。従って、インターフェイスはRS-232Cとほぼ同じ構成になっています。

X1turboでは、シリアルマウスインターフェイスとしてZ80-SIO(Serial Input Output)のBチャンネルを使用しています。(X1ではRS-232C/マウスボードはオプション)パラレルデータとシリアルデータの変換はこのSIOによって行われます。マウスインターフェイスの2番ピン(CONT信号)はSIOのRTSB(チャンネルB送信要求)に、3番ピンはRxDB(チャンネルB受信データ端子)に接続されています。

マウスを使うときは、SIOと、SIOにクロックを与えるCTC(Counter/Timer Circuit)のインシャライズが必要です。マウスからのデータは、ボーレートが4800ボーの非同期(調歩式)のシリアルデータです。そこで、CTCの内部クロック分周比を26、SIOは1/16と設定し、ボーレートを4800ボーにします。

9-3-2 マウスからのデータ

マウスからのデータは、1回につき3バイトです。1バイトは次のようなフィールド構成になっています。

- (1) 未送信状態 H
- (2) スタートビット L(1ビット)
- (3) データ D0~D7(8ビット)
- (4) ストップビット H(1ビット)

マウスから送られてくる3バイトのデータは、次のような構成になっています。

データNO.	内 容
1	ステータス
2	X 方向データ 128～127
3	Y 方向データ 128～127

表9-4 マウスのデータ構成

データ	内 容
D ₁	スイッチ 1 の状態を示します。0 =OFF, 1 =ON
D ₂	スイッチ 2 の状態を示します。0 =OFF, 1 =ON
D ₃	
D ₄	
D ₅	オーバーフロービット, Xが128以上の時 1
D ₅	アンダーフロービット, Xが-129以下の時 1
D ₆	オーバーフロービット, Yが128以上の時 1
D ₇	アンダーフロービット, Yが-129以下の時 1

表9-5 1 バイト目のステータス

9-3-3 データ受信手順

- (1) SIO および CTC をマウスインターフェイス用に設定しておきます。
ボーレート：4800ボー、SIO および CTC で設定
フィールド：8 ビットデータ、パリティなし、1 ストップビット、SIO で設定。
- (2) SIO の RTSB(CONT 信号)をHからLにします。(CONT 信号の立ち下がりが、マウスからのデータ要求信号になります。)ただし、L の状態が 500 μ sec 以上ないとエラーになる場合があります。
- (3) 出力データ応答時間の後、先ほど述べたフォーマットでデータを読み込みます。
- (4) CONT 信号をHに戻します。

9-4 RS-232C インターフェイス

RS-232C インターフェイスとは米国の EIA (Electronic Industries Association) により制定されたデータターミナルとモデムの接続方式の規格です。パソコンの世界では、シリアル入出力の標準インターフェイスとして広く普及しています。

9-4-1 ピン配置

X1turbo は、RS-232C インターフェイスを標準装備しています。このインターフェイスには Z80-SIO の A チャンネルと、Z80-CTC のチャンネル 1 が使用されています。(X1 ではオプション)

標準のRS-232C

ピンNo.	信 号 名	
1	FG(frame ground)	保安用アース
7	SG(signal ground)	信号用アース
2	SD(send data)	送信データ
3	RD(received data)	受信データ
4	RS(request to send)	送信要求
5	CS(clear to send)	受信可
6	DR(data set ready)	モデム・レディ
20	ER(equipment ready)	端末レディ
8	CD(carrier detect)	キャリア検出
22	CI(calling indication)	呼び出し表示

17, RT：受信信号用エレメントタイミング
他はN. C.
X1 では、15, ST₂：送信信号エレメントタイミング 2
24, ST₁：送信信号エレメントタイミング 1
が追加されています。

表9-6 RS-232Cインターフェイスコネクタ端子図

9-4-2 通信モード

X1シリーズのRS-232Cインターフェイスは、Z80-SIOの非同期、同期、SDLCの、どのモードでも使用できるように設計されています。

(1) 非同期：非同期通信では、次のような形式のビットデータを順番に送ることで、1文字分のデータを送信します。

スタートビット：必ず1ビット

データビット：データビットは5, 6, 7, 8ビットのいずれか。

パリティビット：偶数パリティ, 奇数パリティ, パリティ無しのいずれか。

ストップビット：1, 1.5, 2ビットのいずれか。

受信側は、スタートビットを受けたら、時間を測りながら1文字分のデータを取り込んでいきます。

この非同期通信は同期用の特別な信号線が要らないため、マイコン通信によく用いられます。任意の速度の装置を使用できますが、通信の効率はあまりよくありません。

(2) 同期：システム全体に共通したクロック信号で同期をとる方式で、クロックに合わせてデータを転送します。一定のタイミングで動作するので同程度の速度で動作する装置でないと効率が悪くなります。

X1turboではSIOのクロックとして内部クロック、外部クロックのどちらでも使用できます。内部クロック、外部クロックの切り替えはSIOのDTRBによって行われています。

DTRB L：内部クロック

H：外部クロック

外部クロックの場合、ST2は送信用、RTは受信用のクロックとなります。またST1にはシステムクロックをCTC(チャンネル1)で分周したクロックを出力しているので、これを利用することもできます。

9-4-3 ボーレートの設定

ボーレートはSIO及びCTCの設定値で決まります。これらの設定とボーレートの関係は、次の通りです。

ボーレート	SIOの設定値	ボーレート計算値	ボーレート計算値 に一番近いCTC出力	CTCの分周比
9600ボー	1/16	153.6kHz	153.85kHz	13
4800	1/16	76.8kHz	76.92kHz	26
2400	1/16	38.4kHz	38.46kHz	52
//	1/64	153.6kHz	153.85kHz	13
1200	1/16	19.2kHz	19.23kHz	104
//	1/64	76.8kHz	76.92kHz	26
600	1/16	9.62kHz	9.62kHz	208
//	1/64	38.4kHz	38.46kHz	52
300	1/64	19.2kHz	19.23kHz	104
150	1/64	9.62kHz	9.62kHz	208

表9-7 SIO, CTCのボーレート別設定値

9-4-4 モデム

マイコン通信でネットワークを作る場合は、たいてい一般の電話回線を使用します。しかし、一般の電話回線はだいたい 300Hz から 3.4kHz くらいの信号伝達を想定して作られており、RS-232C の信号をそのまま伝送することができません。そこで、アナログ回線である電話回線にデータ伝送をする変調装置 (Modulator) と復調装置 (Demodulator) の両方を兼ね備えたモデム (MODEM) というものがようになります。コンピュータのデータ端末とモデムを接続するインターフェイスが RS-232C です。

RS-232C を、入出力装置間の接続に用いるときには次の点に注意して下さい。

- (1) 余っているピンを他の用途に使うと、トラブルを生じる場合があります。
- (2) 相手側が簡易型 RS-232C インターフェイス (SD, RD, SG, FG の信号線しかない) のときは、次の図のように接続しておかないとインターフェイスが動作しないことがあります。

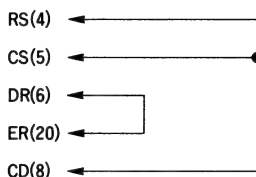


図9-3 相手側が簡易型RS-232Cのときの端末処理

9-5 デジタルテロップ

デジタルテロップは、次のような機能を持つビデオ編集用の周辺機器です。

- (1) テレビ、VTR などの画面とコンピュータ画面を組み合わせて、字幕やコメントなどをいれる (スーパーインポーズ)。
- (2) コンピュータ画面およびスーパーインポーズ画面の VTR への録画
- (3) 外部音声端子からの音声と、コンピュータや映像入力系統からの音声とのサウンドミキシング

X1 シリーズではデジタルテロップとして CZ-8DT が別売されています。この CZ-8DT はジョイスティック用ポート (出力ポートに設定) に、コンピュータコントロール端子を接続することにより

- (1) モードの切り替え
- (2) モニタ出力の選択
- (3) 映像入力を選択
- (4) カットイン／カットアウト機能

をソフトウェアで制御できます。

スーパーインポーズ画面開始は Z80 から 80C49 (サブ CPU) にコードを送ることによって行います。スーパーインポーズ画面ではテレビのコントラストを下げるができます。

CONTROL	ジョイスティック端子 74159入力端子 (テロップ内) FRONT PANEL SW			9	8	7	6	5	4	3	2	1
				NC	GND	NC	\bar{G}_1	\bar{G}_2	D	C	B	A
COMPUTER CONTROL	MONITOR	△	REC OUT	×	GND	×	0		0	0	0	1
		—	MONITOR-3	×	G	×	0		0	0	1	0
		0	MONITOR-2	×	G	×	0		0	0	1	1
		9	MONITOR-1	×	G	×	0		0	1	0	0
	VIDEO IN	8	VIDEO IN-3	×	G	×	0		0	1	0	1
		7	VIDEO IN-2	×	G	×	0		0	1	1	0
		6	VIDEO IN-1	×	G	×	0		0	1	1	1
	MODE	5	VIDEO	×	G	×	0		1	0	0	0
		4	SUPER IMPOSE	×	G	×	0		1	0	0	1
		3	COMPUTER	×	G	×	0		1	0	1	0
	TELOP CUT	2	OUT	×	G	×	0		1	0	1	1
		1	IN	×	G	×	0		1	1	0	0
MANUAL CONTROL	上記のスイッチ操作			×	G	×	1	×	×	×	×	×

表9-8 デジタルテロップのコントロール

なお、ターボZ専用モニタがあれば、モード設定ポート(1FB0H)のビットをON にすることによりインターレススーパーインポーズ(高解像度のスーパーインポーズ)ができます。

サウンドコントロール 画面モードの切り換えと同様にしてチャンネル、音量、パワーON/OFF の設定ができます。

- (1) チャンネルはダイレクトまたはアップ/ダウンの設定
- (2) 音量はアップ/ダウンとミュート(消音)、または一定音量(ノーマル値 42/64 階調)の設定
- (3) パワーON/OFF はトグルまたはダイレクトの設定

9-6 ビデオマルチプロセッサ

ビデオマルチプロセッサは、次のような機能を持つビデオ編集用の周辺機器です。

- (1) 互いに独立な4入力3出力と2入力1出力の2系統のA(audio)/V(visual)スイッチャーによる映像と音声のソースの切り換えができます。この6系統の入力はすべてブリッジ端子で、オーディオ関係はすべてHi-Fiステレオ対応です。
- (2) 映像のカラー信号を安定再生します。(スタビライジング機能)
- (3) カラーバランスの補正を行います。(カラーコレクタ機能)
- (4) 映像の輪郭補正(ビデオエンハンサー機能)
- (5) ABロール編集(複数のビデオを再生しながらの編集)ができるゲンロック端子を持っています。

X1シリーズではビデオマルチプロセッサとしてCZ-8VP1が別売されています。ジョイスティック用ポート(出力ポートに設定)にコンピュータコントロール端子を接続することにより

- (1) A/V スイッチャーの切り換え
- (2) エンハンサーおよびカラーコレクターのON/OFFを行うことができます。

端子 ビデオマルチプロセッサ	コンピュータコントロールコード									T の 値
	9	8	7	6	5	4	3	2	1	
	NC	GND	NC	G2	G1	D	C	B	A	
入カー 1	×	GND	×			0	0	0	1	1
入カー 2	×	G	×			0	0	1	0	2
入カー 3	×	G	×			0	0	1	1	3
入カー 4	×	G	×			0	1	0	0	4
入カー A	×	G	×			0	1	0	1	5
入カー B	×	G	×			0	1	1	0	6
————	×	G	×			0	1	1	1	7
————	×	G	×			1	0	0	0	8
————	×	G	×			1	0	0	1	9
————	×	G	×			1	0	1	0	10
エンハンザON-OFF	×	G	×			1	0	1	1	11
————	×	G	×			1	1	0	0	12
カラーコレクタOF	×	G	×			1	1	0	1	13
カラーコレクタOFF	×	G	×			1	1	1	0	14
————	×	G	×			1	1	1	1	15

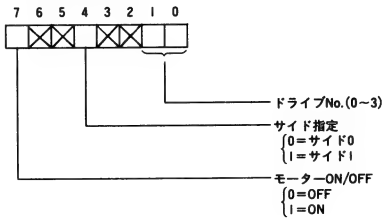
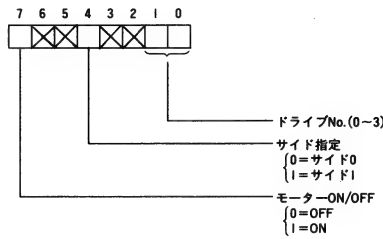
表9-9

付 録

付録 A

I/Oマップ

アドレス	名 前	機 種	IN/OUT	内 容
0700 0701 0704 0705 0706 0707	FM音源/CTC	X/T/Z	OUT IN/OUT IN/OUT IN/OUT IN/OUT IN/OUT	YM2151 アドレスポート YM2151 データポート CTC チャンネル 0 CTC チャンネル 1 CTC チャンネル 2 CTC チャンネル 3 turboZ には CTC は付いていない。 ただし、0704H 番地は FM 音源機能のソフト ウェアチェックに使う。
0800 0801	カラーイメージボード	X/T/Z	OUT IN	カラーイメージボードコントロール 画像データ読み込み
0A00 0A04 0A05 0A06 0A07	立体ボード/CTC	X/T/Z	OUT IN/OUT IN/OUT IN/OUT IN/OUT	立体ボードコントロール CTC チャンネル 0 CTC チャンネル 1 CTC チャンネル 2 CTC チャンネル 3
0B00	増設RAM/ROM バンク切り換え	T/Z	OUT	<p>0800H 7 6 5 4 3 2 1 0 X X X X CS 0~15番のバンクを指定 この1ビットでシステム内かI/Oスロット内かを指定する。 { 0 = I/Oスロット内メモリ 1 = システム内メモリ</p> <p>0000H スロット内0番 スロット内15番 システム内メモリ 7FFFH FFFFH</p>
0C*0 0C*1 0C*2 0C*3 0C*4 0C*5 0C*6 0C*7	RS-232Cカード	X	IN/OUT IN/OUT OUT OUT OUT OUT OUT OUT	データ R/W コントロール、ステータス R/W 送信 IEO をリセット 受信 IEO をリセット 送信割り込み許可 送信割り込み禁止 受信割り込み許可 受信割り込み禁止 I/O アドレス中の * はディップスイッチで設定する。 このポートは、XI 用の RS-232C カードのものであり、turbo 内蔵の RS-232C とは異なる。
0D00 0D01 0D02 0D03	外部RAMボード (EMM)	X/T/Z	OUT OUT OUT IN/OUT	EMM0 の場合 アドレス下位指定 (00H~FFH) アドレス中位指定 (00H~FFH) アドレス上位指定 (00H~04H) データ R/W データのリード/ライトの際、内部アドレスは自動的に加算される。 EMM1 は 0D04H 番地から、EMM2 は 0D08H 番地からと、4 バイトごとに増えて最高64枚までつながる。

アドレス	名 前	機 種	IN/OUT	内 容
0E00 0E01 0E02 0E03 0E80 0E81 0E82 0E80 0E81 0E82	外部ROM BASIC ROM 漢字ROM(CZ8KR) 増設用EPROM	X	OUT OUT OUT IN IN/OUT IN/OUT OUT IN/OUT IN/OUT OUT	アドレス上位指定 アドレス中位指定 アドレス下位指定 データリード 左側データ/アドレス下位指定(00H~FFH) 右側データ/アドレス上位指定(00H~FFH) 00H:増設用EPROMセレクト 01H:漢字ROMセレクト ROM1データ/アドレス(00H~FFH) ROM2データ/アドレス(00H~FFH) 00H:増設用EPROMセレクト 01H:漢字ROMセレクト
0FD1 0FD2 0FD3	ハードディスク	X/T/Z		コントロール コントロール コントロール
0FE8 0FE9 0FEA 0FEB 0FEC 0FED 0FEE 0FEF	8インチFD	T/Z	IN OUT IN/OUT IN/OUT IN/OUT IN OUT IN IN IN	ステータスレジスタ コマンドレジスタ トラックレジスタ セクタレジスタ データレジスタ FM方式指定 ドライブナンバー, ディスクサイド, モーターONレジスタ MFM方式指定 1.6Mタイプ指定 500K/1M切り換え(無意味) 0FECH出力内容の意味 
0FF8 0FF9 0FFA 0FFB 0FFC 0FFD 0FFE 0FFF	5インチFD	X/T/Z	IN OUT IN/OUT IN/OUT IN/OUT IN OUT IN IN IN	ステータスレジスタ コマンドレジスタ トラックレジスタ セクタレジスタ データレジスタ FM方式指定 ドライブナンバー, ディスクサイド, モーターONレジスタ MFM方式指定 1.6M(2HD)タイプ指定 500K(2D)/1M(2DD)切り換え 

アドレス	名前	機種	IN/OUT	内容																																																																																																
1 0 0 0 1 1 0 0 1 2 0 0	グラフィックパレット	X/T/Z Z	OUT IN/OUT	X1/turbo および turboZ のコンパチモード 7 6 5 4 3 2 1 0 10** <table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> 11** <table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> 12** <table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> 縦1列を3桁の2進数とみなして、パレットコードとする。 turboZ の多色モード 1)4096色モード <table border="1"><thead><tr><th></th><th colspan="4">アドレス</th><th colspan="4">データ</th></tr><tr><th></th><th colspan="2">G</th><th colspan="2">R</th><th colspan="2">B</th><th colspan="2">B'</th></tr></thead><tbody><tr><td>10</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>11</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>12</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table> この12ビットでパレットコードを指定 この12ビット(4ビット×3回)でカラーコードを指定																										アドレス				データ					G		R		B		B'		10	7	6	5	4	3	2	1	0										11	7	6	5	4	3	2	1	0										12	7	6	5	4	3	2	1	0									
	アドレス				データ																																																																																															
	G		R		B		B'																																																																																													
10	7	6	5	4	3	2	1	0																																																																																												
11	7	6	5	4	3	2	1	0																																																																																												
12	7	6	5	4	3	2	1	0																																																																																												
1 3 0 0	プライオリティ	X/T/Z	OUT	7 6 5 4 3 2 1 0 <table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> 各々のグラフィック画面のカラーコードを、テキスト画面より優先して表示するかどうか決定する。 たとえば第6ビットが1の場合、黄色はテキスト画面より優先して表示される。																																																																																																
1 4 ** 1 5 ** 1 6 ** 1 7 ** 1 4 *O 1 4 *F 1 5 *O 1 5 *F 1 6 *O 1 6 *F 1 7 *O 1 7 *F	CG, 漢字ROM, PCG アクセス X 通常アクセスモード X turbo 高速アクセスモード	X/T/Z X/T/Z T/Z	IN/OUT IN IN/OUT IN/OUT IN/OUT IN IN/OUT IN/OUT IN/OUT	CGROM アクセス PCG BLUE アクセス PCG RED アクセス PCG GREEN アクセス CG, 漢字ROM アクセス PCG BLUE アクセス PCG RED アクセス PCG GREEN アクセス I/O ポート IFDOH 番地の第5ビットが0のとき通常アクセスモード、1のとき高速アクセスモード																																																																																																
1 8 0 0 1 8 0 1	CRTC	X/T/Z	OUT	CRTC レジスタ NO.の設定(0~17) CRTC レジスタへのデータ(00H~FFH)																																																																																																

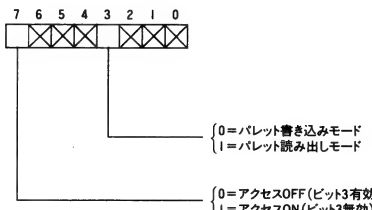
アドレス	名 前	機 種	IN/OUT	内 容																																																																							
1900	サブCPU80C49 (8255①)	X/T/Z	IN/OUT	<table><tr><th>グループ</th><th>ポート端子</th><th>コントロール内容</th><th>アクティブ</th></tr><tr><td rowspan="10">A</td><td>PA₇</td><td rowspan="10">Z80とのデータ入出力 (IN/OUT) (1900H)</td><td>—</td></tr><tr><td>PA₆</td><td>—</td></tr><tr><td>PA₅</td><td>—</td></tr><tr><td>PA₄</td><td>—</td></tr><tr><td>PA₃</td><td>—</td></tr><tr><td>PA₂</td><td>—</td></tr><tr><td>PA₁</td><td>—</td></tr><tr><td>PA₀</td><td>—</td></tr><tr><td>PC₇</td><td>Z80Aに対してデータ受け取り指示信号</td><td>L</td></tr><tr><td>PC₆</td><td>Z80AがポートAからデータ受け取り信号</td><td>L</td></tr><tr><td rowspan="14">B</td><td>PC₅</td><td>Z80Aに対してデータ転送禁止信号</td><td>H</td></tr><tr><td>PC₄</td><td>Z80AからのデータポートBに入力/ラッチ指示信号</td><td>L</td></tr><tr><td>PC₃</td><td>未使用</td><td>—</td></tr><tr><td>PC₂</td><td>カセットLEDの点灯 (H:READ, L:WRITE)</td><td>—</td></tr><tr><td>PC₁</td><td>Z80AへのBREAK信号</td><td>L</td></tr><tr><td>PC₀</td><td>カセットのEJECTソレノイドコントロール</td><td>L</td></tr><tr><td>PB₇</td><td>OBF信号</td><td>—</td></tr><tr><td>PB₆</td><td>ACK信号</td><td>—</td></tr><tr><td>PB₅</td><td>APSS (無記録部検出)</td><td>—</td></tr><tr><td>PB₄</td><td>EJECT SWセンサ</td><td>L</td></tr><tr><td>PB₃</td><td>未使用</td><td>—</td></tr><tr><td>PB₂</td><td>カセットテープの書き込み禁止用の爪がある状態</td><td>H</td></tr><tr><td>PB₁</td><td>カセットがセットされている状態</td><td>H</td></tr><tr><td>PB₀</td><td>テープエンド検出</td><td>L</td></tr></table>	グループ	ポート端子	コントロール内容	アクティブ	A	PA ₇	Z80とのデータ入出力 (IN/OUT) (1900H)	—	PA ₆	—	PA ₅	—	PA ₄	—	PA ₃	—	PA ₂	—	PA ₁	—	PA ₀	—	PC ₇	Z80Aに対してデータ受け取り指示信号	L	PC ₆	Z80AがポートAからデータ受け取り信号	L	B	PC ₅	Z80Aに対してデータ転送禁止信号	H	PC ₄	Z80AからのデータポートBに入力/ラッチ指示信号	L	PC ₃	未使用	—	PC ₂	カセットLEDの点灯 (H:READ, L:WRITE)	—	PC ₁	Z80AへのBREAK信号	L	PC ₀	カセットのEJECTソレノイドコントロール	L	PB ₇	OBF信号	—	PB ₆	ACK信号	—	PB ₅	APSS (無記録部検出)	—	PB ₄	EJECT SWセンサ	L	PB ₃	未使用	—	PB ₂	カセットテープの書き込み禁止用の爪がある状態	H	PB ₁	カセットがセットされている状態	H	PB ₀	テープエンド検出	L
グループ	ポート端子	コントロール内容	アクティブ																																																																								
A	PA ₇	Z80とのデータ入出力 (IN/OUT) (1900H)	—																																																																								
	PA ₆		—																																																																								
	PA ₅		—																																																																								
	PA ₄		—																																																																								
	PA ₃		—																																																																								
	PA ₂		—																																																																								
	PA ₁		—																																																																								
	PA ₀		—																																																																								
	PC ₇		Z80Aに対してデータ受け取り指示信号	L																																																																							
	PC ₆		Z80AがポートAからデータ受け取り信号	L																																																																							
B	PC ₅	Z80Aに対してデータ転送禁止信号	H																																																																								
	PC ₄	Z80AからのデータポートBに入力/ラッチ指示信号	L																																																																								
	PC ₃	未使用	—																																																																								
	PC ₂	カセットLEDの点灯 (H:READ, L:WRITE)	—																																																																								
	PC ₁	Z80AへのBREAK信号	L																																																																								
	PC ₀	カセットのEJECTソレノイドコントロール	L																																																																								
	PB ₇	OBF信号	—																																																																								
	PB ₆	ACK信号	—																																																																								
	PB ₅	APSS (無記録部検出)	—																																																																								
	PB ₄	EJECT SWセンサ	L																																																																								
	PB ₃	未使用	—																																																																								
	PB ₂	カセットテープの書き込み禁止用の爪がある状態	H																																																																								
	PB ₁	カセットがセットされている状態	H																																																																								
	PB ₀	テープエンド検出	L																																																																								
1A*0 ↓ 1A*3	8255②	X/T/Z	IN/OUT	<table><tr><th>ポート</th><th>ポート端子</th><th>コントロール内容</th><th>アクティブ</th></tr><tr><td rowspan="8">A OUT (1A0H)</td><td>PA₇</td><td rowspan="8">プリンター出力データ</td><td>—</td></tr><tr><td>PA₆</td><td>—</td></tr><tr><td>PA₅</td><td>—</td></tr><tr><td>PA₄</td><td>—</td></tr><tr><td>PA₃</td><td>—</td></tr><tr><td>PA₂</td><td>—</td></tr><tr><td>PA₁</td><td>—</td></tr><tr><td>PA₀</td><td>—</td></tr><tr><td rowspan="7">B IN (1A0H)</td><td>PB₇</td><td>垂直帰線期間信号</td><td>L</td></tr><tr><td>PB₆</td><td>データ転送禁止信号</td><td>H</td></tr><tr><td>PB₅</td><td>80C49からのデータ受け取り可能指示信号</td><td>L</td></tr><tr><td>PB₄</td><td>プリンターからの入力可能指示信号</td><td>L</td></tr><tr><td>PB₃</td><td>垂直同期信号</td><td>H</td></tr><tr><td>PB₂</td><td>カセット読み出しデータ</td><td>—</td></tr><tr><td>PB₀</td><td>BREAK信号</td><td>L</td></tr><tr><td rowspan="7">C OUT (1A0H)</td><td>PC₇</td><td>立ち上がりでプリンターは入力データをサンプルする</td><td>立ち上げ</td></tr><tr><td>PC₆</td><td>80/40桁 (1=40桁, 0=80桁)</td><td>—</td></tr><tr><td>PC₅</td><td>立ち下げで同時アクセスモード</td><td>立ち上げ</td></tr><tr><td>PC₄</td><td>スムーズスクロール信号</td><td>L</td></tr><tr><td>PC₃</td><td>—</td><td>—</td></tr><tr><td>PC₂</td><td>—</td><td>—</td></tr><tr><td>PC₀</td><td>カセットテープへの書き込みデータ</td><td>—</td></tr></table> <p>コントロールレジスタ設定ポート OUT (1A03H)</p>	ポート	ポート端子	コントロール内容	アクティブ	A OUT (1A0H)	PA ₇	プリンター出力データ	—	PA ₆	—	PA ₅	—	PA ₄	—	PA ₃	—	PA ₂	—	PA ₁	—	PA ₀	—	B IN (1A0H)	PB ₇	垂直帰線期間信号	L	PB ₆	データ転送禁止信号	H	PB ₅	80C49からのデータ受け取り可能指示信号	L	PB ₄	プリンターからの入力可能指示信号	L	PB ₃	垂直同期信号	H	PB ₂	カセット読み出しデータ	—	PB ₀	BREAK信号	L	C OUT (1A0H)	PC ₇	立ち上がりでプリンターは入力データをサンプルする	立ち上げ	PC ₆	80/40桁 (1=40桁, 0=80桁)	—	PC ₅	立ち下げで同時アクセスモード	立ち上げ	PC ₄	スムーズスクロール信号	L	PC ₃	—	—	PC ₂	—	—	PC ₀	カセットテープへの書き込みデータ	—					
ポート	ポート端子	コントロール内容	アクティブ																																																																								
A OUT (1A0H)	PA ₇	プリンター出力データ	—																																																																								
	PA ₆		—																																																																								
	PA ₅		—																																																																								
	PA ₄		—																																																																								
	PA ₃		—																																																																								
	PA ₂		—																																																																								
	PA ₁		—																																																																								
	PA ₀		—																																																																								
B IN (1A0H)	PB ₇	垂直帰線期間信号	L																																																																								
	PB ₆	データ転送禁止信号	H																																																																								
	PB ₅	80C49からのデータ受け取り可能指示信号	L																																																																								
	PB ₄	プリンターからの入力可能指示信号	L																																																																								
	PB ₃	垂直同期信号	H																																																																								
	PB ₂	カセット読み出しデータ	—																																																																								
	PB ₀	BREAK信号	L																																																																								
C OUT (1A0H)	PC ₇	立ち上がりでプリンターは入力データをサンプルする	立ち上げ																																																																								
	PC ₆	80/40桁 (1=40桁, 0=80桁)	—																																																																								
	PC ₅	立ち下げで同時アクセスモード	立ち上げ																																																																								
	PC ₄	スムーズスクロール信号	L																																																																								
	PC ₃	—	—																																																																								
	PC ₂	—	—																																																																								
	PC ₀	カセットテープへの書き込みデータ	—																																																																								

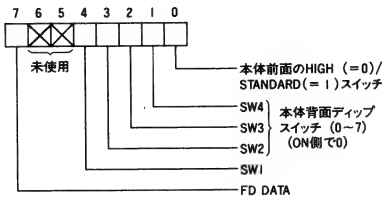
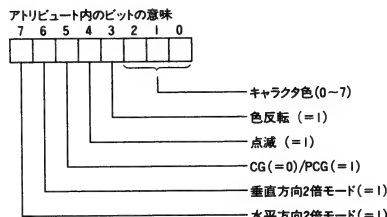
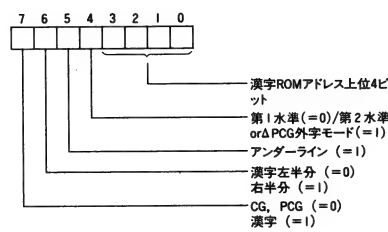
アドレス	名 前	機 種	IN/OUT	内 容																																																																																																														
				<p>8255モード制御(7=1) 1A03H</p> <p>7 6 5 4 3 2 1 0</p> <p>1</p> <table><tr><th colspan="4">グループ制御</th></tr><tr><td>ポ ー ト C</td><td>0</td><td>出</td><td>力</td></tr><tr><td>(下位)</td><td>1</td><td>入</td><td>力</td></tr><tr><td>ポ ー ト B</td><td>0</td><td>出</td><td>力</td></tr><tr><td></td><td>1</td><td>入</td><td>力</td></tr><tr><td>モード選択</td><td>0</td><td>モード0</td><td></td></tr><tr><td></td><td>1</td><td>モード1</td><td></td></tr></table> <table><tr><th colspan="4">グループ制御</th></tr><tr><td>ポ ー ト C</td><td>0</td><td>出</td><td>力</td></tr><tr><td>(上位)</td><td>1</td><td>入</td><td>力</td></tr><tr><td>ポ ー ト A</td><td>0</td><td>出</td><td>力</td></tr><tr><td></td><td>1</td><td>入</td><td>力</td></tr><tr><td>モード選択</td><td>0 0</td><td>モード0</td><td></td></tr><tr><td></td><td>0 1</td><td>モード1</td><td></td></tr><tr><td></td><td>1 X</td><td>モード2</td><td></td></tr></table> <p>8255ビット・セット/リセット(7=0) (ポートCに対して) 1A03H</p> <p>7 6 5 4 3 2 1 0</p> <p>0</p> <p>無効ビット</p> <table><tr><td>ビット・セット/リセット</td><td>0</td><td>リセット</td></tr><tr><td></td><td>1</td><td>セット</td></tr></table> <table><tr><th colspan="8">ポートC ビット選択</th></tr><tr><td>ビット</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>D₁</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>D₂</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>D₃</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table> <p>プリンターへの出力は、次のような手順で行う。</p> <p>① BUSY(PB3)が0になるのを待つ</p> <p>② ポート A (1A00H)へデータを出力する</p> <p>③ PC7 を立ち上げる (*→0→1)</p> <p>PB7(垂直帰線期間信号), PB2(垂直同期信号)は CRT の状態を読み出すビットで、特に PB7は PCG アクセス時に重要である。</p> <p>PB6(データ転送禁止信号)はサブ CPU との交信用に使われている。PB6=1 はメイン CPU の送信禁止を示す。</p> <p>PB1 はカセットからの読み出しデータである。</p> <p>PB0 はサブ CPU からの信号で、CMT がPLAY 中 BREAK キーが押された時などに0になる。</p> <p>PC6 は80/40桁の切り換えに使う。</p> <p>PC5 は立ち下げ (*→1→0)で同時アクセスモードになる。ただし同時アクセスモードに移行する前に DI 命令で割り込みを禁止しておく必要がある。</p> <p>PC4 は0のときに CRTC の5 番レジスタとともに、スーパーインポーズ時のスムーズスクロールを行う。</p> <p>PC0 はカセットテープの書き込みはコマンドであるが、任意の長さの"1"を書けるわけではない。</p> <p>なお、モード設定は IPL が行うので特に改めて設定する必要はない。</p>	グループ制御				ポ ー ト C	0	出	力	(下位)	1	入	力	ポ ー ト B	0	出	力		1	入	力	モード選択	0	モード0			1	モード1		グループ制御				ポ ー ト C	0	出	力	(上位)	1	入	力	ポ ー ト A	0	出	力		1	入	力	モード選択	0 0	モード0			0 1	モード1			1 X	モード2		ビット・セット/リセット	0	リセット		1	セット	ポートC ビット選択								ビット	0	1	2	3	4	5	6	7	D ₁	0	1	0	1	0	1	0	1	D ₂	0	0	1	1	0	0	1	1	D ₃	0	0	0	0	1	1	1	1
グループ制御																																																																																																																		
ポ ー ト C	0	出	力																																																																																																															
(下位)	1	入	力																																																																																																															
ポ ー ト B	0	出	力																																																																																																															
	1	入	力																																																																																																															
モード選択	0	モード0																																																																																																																
	1	モード1																																																																																																																
グループ制御																																																																																																																		
ポ ー ト C	0	出	力																																																																																																															
(上位)	1	入	力																																																																																																															
ポ ー ト A	0	出	力																																																																																																															
	1	入	力																																																																																																															
モード選択	0 0	モード0																																																																																																																
	0 1	モード1																																																																																																																
	1 X	モード2																																																																																																																
ビット・セット/リセット	0	リセット																																																																																																																
	1	セット																																																																																																																
ポートC ビット選択																																																																																																																		
ビット	0	1	2	3	4	5	6	7																																																																																																										
D ₁	0	1	0	1	0	1	0	1																																																																																																										
D ₂	0	0	1	1	0	0	1	1																																																																																																										
D ₃	0	0	0	0	1	1	1	1																																																																																																										

アドレス	名 前	機 種	I N / O U T	内 容
1B** 1C**	PSG ジョイスティック	X/T/Z	I N / O U T O U T	<p>PSG データ (00H~FFH) PSG レジスタ指定 (0~15) PSG (AY-3-8910) へのアクセスに用いるポートである。ジョイスティックにアクセスするには PSG のレジスタ R7, R14, R15 を使う。ジョイスティックからの入力は無論理である。なお、レジスタ番号の指定は一度行えば続けてアクセスする際には再指定する必要はない。</p> <p>ジョイスティックのデータの意味 (R14, R15)</p>
1D** 1E**	IPL (BIOS) ROM ON/OFF	X/T/Z	O U T O U T	<p>IPL ROM ON IPL ROM OFF 出力するデータはダミーである。ON にすると 0000H から 7FFFH までが ROM に切り換わる。そのため、ROM 切り換えを行うときには、それを行う OUT 命令が 8000H 以降になければならない。ROM が ON の時に 0H へジャンプすると IPL が起動する。 また、ROM が ON のときに 0000H から 7FFFH のメモリーにデータを書き込むと、RAM に書き込まれる (ちなみに、読み出しは ROM からである)。この手法はシャドウ RAM と呼ばれ、これにより 64KB が一括してロードできる。</p>
1F8*	DMA	T/Z	I / O	<p>DMA へのコマンド、データ Z80DMA コントローラは (メモリ, I/O) → (メモリ, I/O) 間のデータ転送を高速に行うための LSI である。DMA にはデータ転送のみではなく、サーチ機能もある。また、転送には特定の番地の内容を、ある範囲にコピーする機能もある。</p>
1F90 1F91 1F92 1F93	SIO	X/T/Z	I / O I / O I / O I / O	<p>チャンネル A データ チャンネル B データ チャンネル A 制御 チャンネル B 制御 チャンネル B はマウスにつながっていて、ボーレートは 4800 ボーである。 RS-232C カード、CZ-8RS との互換性はない。CZ-8BM2 上の SIO のアドレスは 1F98H 番地から 1F9BH 番地が割り当てられている。このカードには CTC も入っており、アドレスは 1FA8H から 1FABH となっている。これは CTC がボーレートジェネレータの役目もしているからである。</p>
1FA0 1FA1 1FA2 1FA3	CTC チャンネル 0 チャンネル 1 チャンネル 2 チャンネル 3	X/T/Z	I / O I / O I / O I / O	<p>タイマーモード SIO チャンネル A クロック SIO チャンネル B (マウス) クロック カウンタモード チャンネル 0 の使うクロックは 4MHz で、タイマー周期は 4μsec から 16.384sec までである。チャンネル 1, 2 は 2MHz のクロックを使用している。チャンネル 3 はチャンネル 0 をカウントして、最長タイマーは 4.194sec である。 また、SIO のところで説明したように、1FA8H 番地から 1FABH 番地に、もうひとつ CTC をつけることができる。また、FM 音源ボード、立体ボードなどにも CTC が載る。</p>

アドレス	名 前	機 種	IN / OUT	内 容																		
1FB0	Zモード指定	Z	IN / OUT	<table><tr><th>データ内容</th><th>コントロール</th></tr><tr><td>ビット0</td><td>0=インターレーススーパーインポーズしない 1=インターレーススーパーインポーズする</td></tr><tr><td>ビット1</td><td>無効</td></tr><tr><td>ビット2</td><td>0=画像取り込みの階調ノーマル } (ビット7, 3=1のときのみ有効) 1=画像取り込みの階調反転</td></tr><tr><td>ビット3</td><td>0=画像取り込みをしない } (ビット7=1のときのみ有効) 1=画像取り込みをする</td></tr><tr><td>ビット4</td><td>0=4096色1画面モード指定 } (320×200のときのみ有効) 1=64色2画面モード指定</td></tr><tr><td>ビット5</td><td>無効</td></tr><tr><td>ビット6</td><td>無効</td></tr><tr><td>ビット7</td><td>0=X1/turboコンバチモード 1=多色(turboZ)モード</td></tr></table>	データ内容	コントロール	ビット0	0=インターレーススーパーインポーズしない 1=インターレーススーパーインポーズする	ビット1	無効	ビット2	0=画像取り込みの階調ノーマル } (ビット7, 3=1のときのみ有効) 1=画像取り込みの階調反転	ビット3	0=画像取り込みをしない } (ビット7=1のときのみ有効) 1=画像取り込みをする	ビット4	0=4096色1画面モード指定 } (320×200のときのみ有効) 1=64色2画面モード指定	ビット5	無効	ビット6	無効	ビット7	0=X1/turboコンバチモード 1=多色(turboZ)モード
データ内容	コントロール																					
ビット0	0=インターレーススーパーインポーズしない 1=インターレーススーパーインポーズする																					
ビット1	無効																					
ビット2	0=画像取り込みの階調ノーマル } (ビット7, 3=1のときのみ有効) 1=画像取り込みの階調反転																					
ビット3	0=画像取り込みをしない } (ビット7=1のときのみ有効) 1=画像取り込みをする																					
ビット4	0=4096色1画面モード指定 } (320×200のときのみ有効) 1=64色2画面モード指定																					
ビット5	無効																					
ビット6	無効																					
ビット7	0=X1/turboコンバチモード 1=多色(turboZ)モード																					
1FB9 1FBA 1FBB 1FBC 1FBD 1FBE 1FBF	テキストパレット指定	Z	IN / OUT	コントロール 青のカラーコード 赤のカラーコード マゼンタのカラーコード 緑のカラーコード シアンのカラーコード 黄のカラーコード 白のカラーコード 設定データ <table><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></table> <div><div></div><div>B</div><div>R</div><div>G</div></div> RGB各2ビット=64色を指定できる	7	6	5	4	3	2	1	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
7	6	5	4	3	2	1	0															
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>															
1FC0	Zプライオリティ指定	Z	IN / OUT	<table><tr><th>データ内容</th><th>コントロール</th></tr><tr><td>ビット0</td><td>0, 0=テキストはグラフィックより優先</td></tr><tr><td>ビット1</td><td>0, 1=グラフィックはテキストより優先 1, 0=テキストはグラフィック2面の間に入る 1, 1=未定義</td></tr><tr><td>ビット2</td><td>無効</td></tr><tr><td>ビット3</td><td>0=バンク0はバンク1より優先 1=バンク1はバンク0より優先</td></tr><tr><td>ビット4</td><td>0=バンク0, 1のうち片方だけを表示する 1=バンク0, 1を同時に表示する</td></tr><tr><td>ビット5 ビット6 ビット7</td><td>無効</td></tr></table> ビット3, 4は2画面モード(→1FB0H番地のビット4)のときのみ有効。 * 多色(turboZ)かつ320×200モードでのみ意味のあるポート * ビット4=0のときはビット1=0とみなされる * ビット4=1のときは1FD0Hのビット3は無効	データ内容	コントロール	ビット0	0, 0=テキストはグラフィックより優先	ビット1	0, 1=グラフィックはテキストより優先 1, 0=テキストはグラフィック2面の間に入る 1, 1=未定義	ビット2	無効	ビット3	0=バンク0はバンク1より優先 1=バンク1はバンク0より優先	ビット4	0=バンク0, 1のうち片方だけを表示する 1=バンク0, 1を同時に表示する	ビット5 ビット6 ビット7	無効				
データ内容	コントロール																					
ビット0	0, 0=テキストはグラフィックより優先																					
ビット1	0, 1=グラフィックはテキストより優先 1, 0=テキストはグラフィック2面の間に入る 1, 1=未定義																					
ビット2	無効																					
ビット3	0=バンク0はバンク1より優先 1=バンク1はバンク0より優先																					
ビット4	0=バンク0, 1のうち片方だけを表示する 1=バンク0, 1を同時に表示する																					
ビット5 ビット6 ビット7	無効																					
1FC1	画像取り込み位置補正指定	Z	IN / OUT	<table><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></table> 0~255の補正ドット数を指定する。 * 200ラインモード (1FD0Hのビット0=0) のときのみ有効	7	6	5	4	3	2	1	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
7	6	5	4	3	2	1	0															
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>															

アドレス	名前	機種	IN/OUT	内 容
1FC2	モザイク／量子化取り込み指定	Z	IN/OUT	<div><div><div>7 6 5 4 3 2 1 0</div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div><div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div><div>0 0 0 0 1 1 1 1</div></div></div></div>

アドレス	名 前	機 種	IN/OUT	内 容																		
1FC5	多色モードでのグラフィックパレット制御指定	Z	IN/OUT	<p>(このモードは多色モードでのみ有効。10 * * H 番地 - 12 * * H 番地と組み合わせて使う)</p> 																		
1FD0	画面管理	T/Z	(IN)/OUT	<table border="1"> <thead> <tr> <th>データ内容</th><th>コントロール</th></tr> </thead> <tbody> <tr> <td>ビット0</td><td>0 = 低解像度モニター (200ライン) 1 = 高解像度モニター (400ライン) モニター切り換え</td></tr> <tr> <td>ビット1</td><td>0 = 1本ラスタ/ドット 1 = 2本ラスタ/ドット</td></tr> <tr> <td>ビット2</td><td>0 = ノーマル (8 ラスタ/CHAR) (25行, 20行) 1 = 漢字 (16ラスタ/CHAR) (12行, 10行)</td></tr> <tr> <td>ビット3</td><td>0 = バンク 0 表示 1 = バンク 1 表示</td></tr> <tr> <td>ビット4</td><td>0 = バンク 0 アクセス 1 = バンク 1 アクセス</td></tr> <tr> <td>ビット5</td><td>0 = PCG コンパチアクセス 1 = PCG 高速アクセス</td></tr> <tr> <td>ビット6</td><td>0 = 8 ラスタCG アクセス 1 = 16ラスタCG アクセス</td></tr> <tr> <td>ビット7</td><td>0 = アンダーラインなし 1 = アンダーラインあり</td></tr> </tbody> </table> <p>BASIC 起動直後はビット1 = 1となっている。これによりグラフィックを高解像度モニターで200ラインとして扱える(キャラクタは400ライン)。turbo では、グラフィックスが400ラインモードの時には偶数段目がバンク0、奇数段目にバンク1の内容が表示される。ビット1が1になっている時には、その時表示されているバンク(ビット3で指定)の内容のパターンを下の段にも表示する。CRTC ビット2はテキストの25(or20) → 12(or10)行の指定に使う。CRTC の設定と一緒に操作しないと表示が乱れるので注意が必要。ビット3は640×400モードの時は、無意味である。ビット1が0なら400ラインの表示を行う。ビット5はPCGのアクセス方法の指定で、PCGにアクセスするには設定が必要である。ビット6は2種類あるCG-ROMのどちらを読み込むかの指定である。ビット7はアンダーラインの設定である。</p>	データ内容	コントロール	ビット0	0 = 低解像度モニター (200ライン) 1 = 高解像度モニター (400ライン) モニター切り換え	ビット1	0 = 1本ラスタ/ドット 1 = 2本ラスタ/ドット	ビット2	0 = ノーマル (8 ラスタ/CHAR) (25行, 20行) 1 = 漢字 (16ラスタ/CHAR) (12行, 10行)	ビット3	0 = バンク 0 表示 1 = バンク 1 表示	ビット4	0 = バンク 0 アクセス 1 = バンク 1 アクセス	ビット5	0 = PCG コンパチアクセス 1 = PCG 高速アクセス	ビット6	0 = 8 ラスタCG アクセス 1 = 16ラスタCG アクセス	ビット7	0 = アンダーラインなし 1 = アンダーラインあり
データ内容	コントロール																					
ビット0	0 = 低解像度モニター (200ライン) 1 = 高解像度モニター (400ライン) モニター切り換え																					
ビット1	0 = 1本ラスタ/ドット 1 = 2本ラスタ/ドット																					
ビット2	0 = ノーマル (8 ラスタ/CHAR) (25行, 20行) 1 = 漢字 (16ラスタ/CHAR) (12行, 10行)																					
ビット3	0 = バンク 0 表示 1 = バンク 1 表示																					
ビット4	0 = バンク 0 アクセス 1 = バンク 1 アクセス																					
ビット5	0 = PCG コンパチアクセス 1 = PCG 高速アクセス																					
ビット6	0 = 8 ラスタCG アクセス 1 = 16ラスタCG アクセス																					
ビット7	0 = アンダーラインなし 1 = アンダーラインあり																					
1FE0	黒色制御 (スーパーインポーズ時の黒抜き指定)	T/Z	(IN)/OUT	<table border="1"> <thead> <tr> <th>データ内容</th><th>コントロール</th></tr> </thead> <tbody> <tr> <td>ビット0</td><td rowspan="4">} 黒変換するテキストの色を指定 (0-7)</td></tr> <tr> <td>ビット1</td></tr> <tr> <td>ビット2</td></tr> <tr> <td>ビット3</td></tr> <tr> <td>ビット3</td><td>テキストの黒変換のON/OFF</td></tr> <tr> <td>ビット4</td><td>グラフィックの黒(透明)を黒変換</td></tr> <tr> <td>ビット5</td><td>グラフィックの青を黒変換</td></tr> <tr> <td>ビット6</td><td>ブランキング期間(枠)を黒変換</td></tr> <tr> <td>ビット7</td><td>未使用</td></tr> </tbody> </table> <p>(1 = ON, 0 = OFF)</p> <p>グラフィックを黒変換するためにはパレットが0になっている必要がある。また、このポートはturbo ではOUTのみであるがZではINも可能。</p>	データ内容	コントロール	ビット0	} 黒変換するテキストの色を指定 (0-7)	ビット1	ビット2	ビット3	ビット3	テキストの黒変換のON/OFF	ビット4	グラフィックの黒(透明)を黒変換	ビット5	グラフィックの青を黒変換	ビット6	ブランキング期間(枠)を黒変換	ビット7	未使用	
データ内容	コントロール																					
ビット0	} 黒変換するテキストの色を指定 (0-7)																					
ビット1																						
ビット2																						
ビット3																						
ビット3	テキストの黒変換のON/OFF																					
ビット4	グラフィックの黒(透明)を黒変換																					
ビット5	グラフィックの青を黒変換																					
ビット6	ブランキング期間(枠)を黒変換																					
ビット7	未使用																					

アドレス	名 前	機 種	I N / O U T	内 容																																																																								
1 F F 0	スタートポート	T/Z	I N	<p>IPL などのプログラムが必要な情報を得るためのフラグである。</p>  <p>ビット 1 ~ 3 で BOOT 時のディスクを指定する。SW の番号とビット順が逆である事に注意が必要である。以下にディスクの種類の表を示す。</p> <table border="1"> <thead> <tr> <th>No.</th> <th>SW2</th> <th>SW3</th> <th>SW4</th> <th>セレクト</th> <th>容 量</th> <th>記録方式</th> <th>フォーマット</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>5(3)インチ</td> <td>320Kバイト</td> <td>2D : 両面倍密度</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>5(3)インチ</td> <td>640Kバイト</td> <td>2DD : 両面倍密度倍トラック</td> <td></td> </tr> <tr> <td>2</td> <td>0</td> <td>1</td> <td>0</td> <td>5インチ</td> <td>1Mバイト</td> <td>2HD : 両面高密度</td> <td></td> </tr> <tr> <td>3</td> <td>0</td> <td>1</td> <td>1</td> <td>5インチ</td> <td>1Mバイト</td> <td>2HD : 両面高密度</td> <td>IBM</td> </tr> <tr> <td>4</td> <td>1</td> <td>0</td> <td>0</td> <td>8インチ</td> <td>1Mバイト</td> <td>2D256: 両面倍密度</td> <td></td> </tr> <tr> <td>5</td> <td>1</td> <td>0</td> <td>1</td> <td>8インチ</td> <td>1Mバイト</td> <td>2D256: 両面倍密度</td> <td>IBM</td> </tr> <tr> <td>6</td> <td>1</td> <td>1</td> <td>0</td> <td>8インチ</td> <td>240Kバイト</td> <td>1S128: 片面単密度</td> <td>IBM</td> </tr> <tr> <td>7</td> <td>1</td> <td>1</td> <td>1</td> <td>ハードディスク</td> <td>10Mバイト</td> <td></td> <td></td> </tr> </tbody> </table>	No.	SW2	SW3	SW4	セレクト	容 量	記録方式	フォーマット	0	0	0	0	5(3)インチ	320Kバイト	2D : 両面倍密度		1	0	0	1	5(3)インチ	640Kバイト	2DD : 両面倍密度倍トラック		2	0	1	0	5インチ	1Mバイト	2HD : 両面高密度		3	0	1	1	5インチ	1Mバイト	2HD : 両面高密度	IBM	4	1	0	0	8インチ	1Mバイト	2D256: 両面倍密度		5	1	0	1	8インチ	1Mバイト	2D256: 両面倍密度	IBM	6	1	1	0	8インチ	240Kバイト	1S128: 片面単密度	IBM	7	1	1	1	ハードディスク	10Mバイト		
No.	SW2	SW3	SW4	セレクト	容 量	記録方式	フォーマット																																																																					
0	0	0	0	5(3)インチ	320Kバイト	2D : 両面倍密度																																																																						
1	0	0	1	5(3)インチ	640Kバイト	2DD : 両面倍密度倍トラック																																																																						
2	0	1	0	5インチ	1Mバイト	2HD : 両面高密度																																																																						
3	0	1	1	5インチ	1Mバイト	2HD : 両面高密度	IBM																																																																					
4	1	0	0	8インチ	1Mバイト	2D256: 両面倍密度																																																																						
5	1	0	1	8インチ	1Mバイト	2D256: 両面倍密度	IBM																																																																					
6	1	1	0	8インチ	240Kバイト	1S128: 片面単密度	IBM																																																																					
7	1	1	1	ハードディスク	10Mバイト																																																																							
2 0 0 0 ~ 2 7 F F	テキストアトリビュート	X/T/Z	I N / O U T	<p>アトリビュート内のビットの意味</p> 																																																																								
3 0 0 0 ~ 3 7 F F	テキスト V-RAM	X/T/Z	I N / O U T	<p>それぞれのアドレスに画面に対応したキャラクターコードが格納される。また、turbo で漢字を表示する場合は、漢字 ROM アドレスの下位 8 ビットが格納される。漢字コードの場合、漢字の左側 8 ドットか右側 8 ドットかは、漢字 V-RAM によって指定されるので、テキスト V-RAM では設定の必要はない。</p>																																																																								
3 8 0 0 ~ 3 F F F	漢字 V-RAM	T/Z	I N / O U T																																																																									

アドレス	名 前	機 種	IN／OUT	内 容																															
				<p>テキストアトリビュートのビット5と、この漢字V-RAMを使って表示する文字の種類を指定する。その関連を下表に示す。</p> <table><tr><th>テキストアトリビュート</th><th colspan="2">漢字VRAM</th><th rowspan="2">表 示</th></tr><tr><th>ビット5=ROM/RAM</th><th>ビット7=CG/漢字</th><th>ビット4=T/2水準</th></tr><tr><td>0</td><td>0</td><td>無関係</td><td>CG</td></tr><tr><td>0</td><td>1</td><td>0</td><td>漢字(第1水準)</td></tr><tr><td>0</td><td>1</td><td>1</td><td>漢字(第2水準)</td></tr><tr><td>1</td><td>0</td><td>0</td><td>PCG(ノーマル)</td></tr><tr><td>1</td><td>0</td><td>1</td><td>PCG外字①</td></tr><tr><td>1</td><td>1</td><td>無関係</td><td>PCG外字②</td></tr></table>	テキストアトリビュート	漢字VRAM		表 示	ビット5=ROM/RAM	ビット7=CG/漢字	ビット4=T/2水準	0	0	無関係	CG	0	1	0	漢字(第1水準)	0	1	1	漢字(第2水準)	1	0	0	PCG(ノーマル)	1	0	1	PCG外字①	1	1	無関係	PCG外字②
テキストアトリビュート	漢字VRAM		表 示																																
ビット5=ROM/RAM	ビット7=CG/漢字	ビット4=T/2水準																																	
0	0	無関係	CG																																
0	1	0	漢字(第1水準)																																
0	1	1	漢字(第2水準)																																
1	0	0	PCG(ノーマル)																																
1	0	1	PCG外字①																																
1	1	無関係	PCG外字②																																
4 0 0 0～ F F F F	グラフィックV-RAM	X/T/Z	IN／OUT	<div><div><div>ノーマルモード</div><div>0000H</div><div><div></div><div></div><div></div><div></div></div><div>4000H</div><div>B</div><div>8000H</div><div>R</div><div>C000H</div><div>G</div><div>FFFFH</div></div><div><div>同時アクセスモード</div><div>B・R・G</div><div>R・G</div><div>B・G</div><div>B・R</div></div></div> <p>〔備考〕 同時アクセスモードは1A02H番地、turboでのバンク切り換えは1FD0H番地。 グラフィック画面上のドットのアドレスを計算するためには、次の式を用いるとよい。 X=X座標、Y=Y座標として640×200のとき &H4000+(X÷8)+((Y AND 7)*2¹¹)+(Y÷8)*80 320×200の場合は最後の*80が*40になる。 turboの400ラインではバンク0とバンク1が一段おきに表示されているので、アドレス計算はY AND 1をバンクNoとし、Y=Y÷2として前述の式で計算することによって行う。 また、Zでは4096色モード、64色モードが加わり、その色はページ0、1、2、3の順に薄くなっている。</p>																															

付録 B

turboシリーズ

B-1 BIOS-ROMマップ

B-1-1 項目別

80C49, キー入力関係

\$ 0 3 8 A	IPL のキー入力	0 3 8 A
K V E C I N	キー入力のインターラプトモードセット	1 0 D 0
K V E C 0 0	キー入力のノンインターラプトモードセット	1 0 D 6
S 4 9 R E S	コミュニケーションバッファのクリア	1 3 E 5
I N 4 9 S B	データの1バイト入力	1 4 0 8
O T 4 9 S B	データの1バイト出力	1 4 1 3
C O M O U T	コマンドの出力	1 4 3 2
T A K 4 9 S	Z-80A 間でのデータの受け渡し	1 4 3 B
I N K E Y S	キーボードからの1文字入力	1 F F 0
B R K C K S	SHIFT+BREAK もしくは CTRL-C が押されているかのチェック	2 0 D 5
K E Y S N S	入力されたデータが有効かどうかのチェック	2 0 E B
K E Y S N 1	ROM 用の KEYSNS サブルーチン	2 0 F 5
K E Y R A M	キー割り込み時の処理	F 8 4 3

AY-3-8910 関係 BIOS

PSG 関係

P S G I N T	PSG データのイニシャライズ	1 0 B 4
B E E P	BEEP 音の出力(=CTRL-G)	1 B 4 1
T E M P S B	テンポの設定	6 5 6 E
M U S I C S	音楽の演奏	6 5 A C
M U B F S T	音楽データのインターラプトジョブ用データへの変換	6 5 F 2

ジョイスティック関係

S T R I G S	ジョイスティックのトリガーまたはスペースキーの状態チェック	1 D 8 9
S T I C K S	ジョイスティックの方向, もしくはテンキーの値のチェック	1 D 9 2

PCG 関係

C G S E T	PCG の定義	3 2 A D
C G R E A D	CG のパターンデータの読み込み	3 3 0 D

SIO(RS-232C, マウス)関係

SIOCTC	CTC と SIO のイニシャライズ	6 D 3 F
RSINIT	SIO のチャンネルAのモードセット	6 D A 5
RXINP	RS-232C からのデータの入力	6 E 5 9
RXSNS	RS-232C からのデータ入力が可能かのチェック	6 E 8 3
TXOUT	RS-232C へのデータ出力	6 E 8 A
TXSNS	RS-232C へのデータ出力が可能かのチェック	6 E A 7
MOUSE 0	マウス割り込みモードの解除	6 E A F
MOUSE 1	マウス割り込みモードの設定	6 E C 0

かな漢字変換関係

OPENF 9	ユーザー辞書モードのオープン処理	F 8 B 7
OPENF 8	システム辞書モードのオープン処理	F 8 B A
OPENF 7	音訓辞書モードのオープン処理	F 8 B D
FINDF 7	かな漢字変換処理	F 8 C 0
NEXTF 7	次候補漢字のバッファ設定	F 8 C 3
BACKF 7	前候補漢字のバッファ設定	F 8 C 6
X1CLF 7	漢字の選択と学習機能処理	F 8 C 9
NEXTJS	音訓辞書モード次候補, シフト JIS コード入力の処理	F 8 C C
KEYSNN	キーセンスの処理	F 8 E 7

画面表示関係

\$ 0 3 C B	IPL のメッセージ出力	0 3 C B
\$ 0 3 D 9	IPL の 1 文字表示	0 3 D 9
BADSDM	バッドスクリーンモードエラー(コード 25)を返す	1 0 1 D
SCRNSB	スクリーンモードの設定	1 0 D F
CR 4 0 0 S	CRTC の 400 ライン設定	1 1 D 8
ROMASK	BASIC の A S K コマンドの処理	1 1 E 7
WIDTH 8 0	WIDTH80 の設定	1 2 2 0
WIDTH 4 0	WIDTH40 の設定	1 2 2 7
CTRLD 2	コンソールをイニシャライズして SCREEN 0,0 を行う	1 2 B 9
SCRNOT	スクリーンのディスプレイモードの設定	1 2 D B
SCRNIN	スクリーンのアクセスモードの設定	1 3 0 7
STCLST	テキスト V-RAM のクリアー	1 3 7 7
STCLSG	グラフィック V-RAM のクリアー	1 3 9 A
INTCRT	スクリーンのイニシャライズ	1 4 B F
CR TCR 1	CSIZE サポートの CR+LF 出力	1 6 C 5
CR TACC	CSIZE サポートの 1 文字出力	1 6 D 3
DEPRT	画面への文字列出力	1 7 5 4
CR 2	現在のカーソル X 座標が 0 以外の時, CR1 へ飛び改行動作	1 7 7 0

CR1	改行する	1778
TABPRT	HTAB PRINT の処理	1780
SPPRT	画面にスペースを表示	178F
ACCPRT	画面に1文字表示	1791
ACCDIS	画面に1文字表示	179D
TBCALC	テキスト V-RAM 上の任意の行のコネクトフラグのアドレス出力	18B1
ADRCA2	現在のカーソル位置のテキスト V-RAM 上でのアドレスの計算	18BC
CTRLJB	コントロールコード(00H~1FH)出力の処理	18E1
BINPUT	BASIC の INPUT 文と同じ処理を行う	1DC2
INPUTF	BASIC の LINPUT 文と同じ処理を行う	1DE4
BCUYST	HにY座標を入力し、その行の始まっているY座標をDに返す	1F16
ECUYST	HにY座標を入力し、その行の終わり+1のY座標をDに返す	1F25
SCRGET	BASIC の SCRNS\$と同じ働きをする	1F8F
X1HPDS	XFER モードの設定	274E
FKYDS1	ファンクションキーの表示	2A1B
FKYDSS	ファンクションキーモードを表示	2A22
EDLNDS	XFER /ファンクションモードの表示	2A6B
BOXFUL	4角形を描きその内部を塗りつぶす	5507
BOXSUB	4角形を描く	5604
LINESB	直線を引く	569F
ELHPUT	PUT のルーチン	578D
ELHGET	GET のルーチン	57AA
PSETSB	PSET のルーチン	57F1
RESETS	RESET のルーチン	580C
POINTS	A=POINT(HL, DE)	58BD
GRAADR	グラフィックアドレスの算出ルーチン(ウィンドウのチェック付き)	5907
GRAAD2	グラフィックアドレスの算出ルーチン	590F
UPADR	グラフィックアドレスを1ライン分上げる	59A8
DWADR	グラフィックアドレスを1ライン分下げる	59FC
CLSGRA	グラフィック画面のクリア	5A4D
WINDOI	ウィンドウを最大にする	5AD8
WINDST	パラメータを与えてウィンドウを設定	5AEA
TILCOL	タイルバッファにカラーパターンを設定	5B99
HPAINT	任意の部分を指定したカラーでペイント	5EA1
TILSET	タイルバッファにタイルパターンを設定	61A5
PATSUB	PATTERN 処理ルーチン	623D
POLYSB	多角形、または円・弧を描く	630B
SCRRAM	テキスト V-RAM スクロールの処理	F8EA
SETRES	PSET, PRESET, XOR の処理	FA39
SETMD	PSET, PRESET, XOR, POINT1 の処理	FA3D
RESMD	PSET, PRESET, XOR, POINT0 の処理	FA40

漢字処理関係

SFTKTN	シフト JIS コード→区点コードの変換	2 F 0 7
KTNSFT	区点コード→シフト JIS コードの変換	2 F 2 C
JISSFT	JIS 漢字コード→シフト JIS コードの変換	2 F 5 2
SFTJIS	シフト JIS コード→JIS 漢字コードの変換	2 F 8 1
JISVRM	JIS 漢字コード→V-RAM データの変換	2 F B 6
VRMJIS	V-RAM データ→JIS 漢字コードの変換	3 0 3 7
SFTCHK	シフト JIS コードの上位1バイトかのチェック	3 0 9 9
KANDAK	JIS 漢字コードを濁点付きのコードに変換	3 0 A 3
KANHAN	JIS 漢字コードを半濁点付きのコードに変換	3 0 F 2
ASCKAN	アスキーコードを JIS 漢字コードに変換	3 1 1 9
KANASC	JIS 漢字コードをアスキーコードに変換	3 1 A 6

関数関係

IFCALL	イリーガルファンクションコールエラー(コード5)を返す	1 0 0 E
OVERFL	オーバーフローエラー(コード6)を返す	1 0 1 1
DVBYZR	ディビジョンバイゼロエラー(コード11)を返す	1 0 1 4
TYPEMS	タイプミスマッチエラー(コード13)を返す	1 0 1 7
TOOCMP	トゥーコンプレックスエラー(コード16)を返す	1 0 1 A
SUB	$[HL] = [HL] - [DE]$	3 A F 8
ADD	$[HL] = [HL] + [DE]$	3 A F B
CMP	$[HL]$, $[DE]$ の比較	3 D B A
MUL	$[HL] = [HL] \times [DE]$	3 E 0 1
DIV	$[HL] = [HL] \div [DE]$	4 0 3 E
INTDVS	符号付き整数の除算($DE \div HL = DE \cdots HL$)	4 0 E 3
INTDVN	符号無し整数の除算($HLDE \div BC = DE \cdots HL$)	4 1 1 D
INTDVV	符号無し整数の除算($HLDE \div BC = DE \cdots HL$ $HL < BC$)	4 1 2 2
CVFLAS	アスキー文字列の浮動小数点型データへの変換	4 3 5 3
ANDBOH	アスキー文字列の整数型データへの変換	4 4 9 4
CHCKHX	Aレジスタの値が16進数を表すアスキーコードかどうかをチェック	4 4 E 7
CVHLAS	Aレジスタにタイプを入力し、数値を表す文字列を数値に変換する	4 4 F 5
HEXCUL	10進数を表すアスキー文字列を数値に変換する	4 4 F A
TOGLE	$[HL] = -[HL]$	4 5 2 6
MULTEN	$[HL] = [HL] \times 10$	4 5 6 5
DIVTEN	$[HL] = [HL] \div 10$	4 5 7 2
MULDEC	$[HL] = [HL] + A$	4 5 7 F
FLTHX	整数型データ→浮動小数点型データの変換	4 5 A 6
CVNMFL	浮動小数点型データ→符号付きアスキー文字列の変換	4 5 D 2
CVASFL	浮動小数点型データ→符号無しアスキー文字列の変換	4 5 F 3
CVASIN	整数型データ→符号無しアスキー文字列の変換	4 6 A E

CVASII	HLに入っている整数型データ→アスキー文字列の変換	4 6 B 8
CVASSN	整数型データ→符号付きアスキー文字列の変換	4 6 C A
ASC FIV	整数型データ→符号無しアスキー文字列の変換	4 6 E 7
HEXHL0	HLに入っている整数型データ→16進数を表すアスキー文字列の変換	4 6 F I
BINFL0	HLに入っている整数型データ→2進数を表すアスキー文字列の変換	4 6 F B
OCTHL0	HLに入っている整数型データ→8進数を表すアスキー文字列の変換	4 7 0 5
ASCHL	HLに入っている整数型データ→10進数を表すアスキー文字列の変換	4 7 1 5
BINHL	HLに入っている整数型データ→2進数を表すアスキー文字列の変換	4 7 4 7
OCTHL	HLに入っている整数型データ→8進数を表すアスキー文字列の変換	4 7 5 6
KT NHL	HLに入っているシフト JIS コード→区点コードを表すアスキー文字列の変換	4 7 6 F
JISHL	HLに入っているシフト JIS コード→JIS 漢字コードを表すアスキー文字列の変換	4 7 7 5
HEXHLB	DEに入っている整数型データ→16進数を表す符号無しアスキー文字列の変換	4 7 7 9
HEXHL	HLに入っている整数型データ→16進数を表すアスキー文字列に変換後、DEで指定したアドレスに格納する	4 7 7 D
HEXA	Aに入っている HEX データ→16進数を表すアスキー文字列に変換後、DEで指定したアドレスから格納する	4 7 8 A
USNGCV	書式指定による浮動小数点型データ→アスキー文字列の変換を行う	4 9 0 8
HEXFLT	HLで示されたアドレスからの浮動小数点型データが-32768< [HL] <65535 であれば、整数型に変換した後HLレジスタに格納	4 A 6 E
HLFLT	HLで示されたアドレスからの浮動小数点型データが-32768< [HL] <65535 であれば、整数型に変換した後HLレジスタに格納	4 A 7 B
HLFLT0	HLで示されたアドレスからの浮動小数点型データが-32768< [HL] <32767 であれば、整数型に変換した後HLレジスタに格納	4 A 8 2
POWERS	[HL] = [HL] [DE]	4 A D 9
ABS	[HL] =ABS [HL]	4 B 8 2
INTOPR	[HL] =INT] ¼[HL]	4 B 8 A
SQR	[HL] =SQR [HL]	4 B A E
SUM	[HL] =SUM [HL]	4 B C 3
FACG	[HL] =FAC [HL]	4 B F I
ATN	[HL] =ATN [HL]	4 C 3 E
COS	[HL] =COS [HL]	4 D 0 7
SIN	[HL] =SIN [HL]	4 D 2 0
TAN	[HL] =TAN [HL]	4 E 2 5

SGN	[HL] =SGN [HL]	4 E 5 C
RAD	[HL] =RAD [HL]	4 E 8 4
PAI	[HL] =PAI [HL]	4 E 8 D
RND	[HL] =RND [HL]	4 E 9 6
EXP	[HL] =EXP [HL]	4 E C 5
LOG	[HL] =LOG [HL]	4 F D 8
CSNGP	[HL] =CSNG [HL]	5 0 B 0
CDBL	[HL] =CDBL [HL]	5 1 0 2
CSNG	[HL] =CSNG [HL]	5 1 3 1
CINT0	[HL] =CINT [HL]	5 1 6 7
CINT	[HL] =CINT [HL]	5 1 7 9
FIX	[HL] =FIX [HL]	5 1 B E
FIXFLT	[HL] =FIX [HL]	5 1 C 4
FRAC	[HL] =FRAC [HL]	5 2 5 8

データレコーダ関係

TAK49S	SUB CPU(80C49)と Z-80A 間でのデータの受け渡し	1 4 3 B
FMPRHL	CR1 を行った後「Found」ファイルネーム，拡張子」か「Writing」フ ァイルネーム，拡張子」の表示を行う	3 9 D 6
WRTMES	「Writing」のメッセージデータテーブル	3 A D 2
FINMES	「Found」のメッセージデータテーブル	3 A D A
SKPMES	「Skip」のメッセージデータテーブル	3 A E 2
SAVE1	データレコーダへの FBC の出力	7 0 2 0
SAVE2	データレコーダへのデータの出力	7 0 2 4
LOAD1	データレコーダからの FCB の入力	7 0 4 7
LOAD2	データレコーダからのデータの入力	7 0 4 B
VERFY2	データレコーダへ出力したデータとメモリーの内容の比較	7 0 5 C
CMTCOM	データレコーダへコントロールコードを出力	7 2 C 3
CMTSNS	データレコーダの状態チェック	7 2 C D

時計・カレンダー関係

DAYBUF	曜日のメッセージ用データテーブル	0 E 8 8
TAK49S	SUB CPU(80C49)と Z-80A 間のデータの受け渡し	1 4 3 B
CVDATS	日付の読み出しと，アスキー文字列での格納	5 2 9 6
CVDATE	HLで示したアドレスからの3バイトの年・月・日内部コード読み 出しと，DEで示すアドレスからのアスキー文字列での格納	5 2 9 9
CVDAYS	曜日の読み出しと，アスキー文字列での格納	5 2 D F
CVDAY	HLで示したアドレスからの3バイトの年・月・日内部コード読み 出しと，DEで示すアドレスからのアスキー文字列での格納	5 2 E 2
CVT1\$S	時間の読み出しと，アスキー文字列での格納	5 2 F B

CVTIME	HLで示したアドレスからの3バイトの時・分・秒内部コード読み出しと、DEで示したアドレスからのアスキー文字列での格納	5 3 0 0
CVTIMS	BASICのTIME用の秒数の読み出しと、DEで示したアドレスからの格納	5 3 1 6
DATSTS	日付設定	5 3 2 B
DAYSTS	曜日設定	5 3 A 8
T1\$STS	時刻設定	5 3 D 7
T1STS	時刻設定 (TIME = ? CR と同じ)	5 4 1 8

パレット機能関係

PALETI	パレットのイニシャライズとプライオリティの設定	1 3 4 C
STPRIO	パレットとプライオリティの設定	1 3 5 9
PALETF	パレットとプライオリティを全て0にする	1 3 6 C
PALSET	パレットのカラーを設定	1 4 8 0

プリンター関係

PNORDY	プリンターオフラインエラー(コード 73)を返す	1 0 2 9
CR1PRP	FILOUT = 0 → CR1, FILOUT = 1 → CR1LPL へと処理を渡す	3 7 A B
CR1LPL	プリンターへの CR / LF のコード出力と LPOS のクリア	3 7 B 2
CR1LTP	プリンターへの CR / LF のコードを出力	3 8 0 F
ACCPRP	FILOUT = 0 → ACCPRT, FILOUT = 1 → ACCLPL へと処理を渡す	3 8 3 1
ACCLPL	プリンターへの 1 文字出力	3 8 3 9
HLLPRT	HLレジスタの示すアドレスから始まるデータテーブルのデータのプリンターへの出力	3 9 2 7
ACCLPT	プリンターへの 1 文字出力	3 9 8 3
LPTSNS	プリンター状態のチェック	3 9 A 1
TABPRP	FILOUT = 0 → TABPRT, FILOUT = 1 → TABLPL へと処理を渡す	3 9 B A
TABLPL	プリンター水平タブ出力	3 9 C 1
BITDES	ビットイメージ LPRINT 用バッファの出力	F 8 D E
HCOPYS	HCOPY の処理	F 8 E 1
CPSM23	HCOPY の処理 (WIDTH, 20 or WIDTH, 10)	F 8 E 4

フロッピー関係

\$00F5	IPL が正常に BOOT できなかった場合のエラー処理	0 0 F 5
\$021A	IPL からの 2D 3 インチまたは 5 インチディスクのリード	0 2 1 A
DIOERR	デバイス I/O エラー(コード 56)を返す	1 0 0 0
TWRTPR	書き込み禁止エラー(コード 72)を返す	1 0 0 3

DEVUNA	デバイスオフラインエラー(コード 73)を返す	1 0 0 6
FOVER	フォーマットオーバーエラー(コード 36)を返す	1 0 2 0
BADFDC	バッドファイルディスクリプターエラー(コード 65)を返す	1 0 2 3
BADREC	バッドレコードエラー(コード 66)を返す	1 0 2 6
BADFMD	バッドファイルモードエラー(コード 30)を返す	1 0 2 E
BADPAS	パスワード無しエラー(コード 67)を返す	1 0 3 1
FDCRED	デバイスからのデータ入力	7 3 9 D
FDCWRT	デバイスへのデータ出力	7 3 A A
FDCVfy	デバイスのベリファイ	7 3 B 7
DSKRED	DMA を使用しないで 3 インチ or 5 インチディスクからデータの 入力を行う	7 6 C A
DSKWRT	DMA を使用しないで 3 インチ or 5 インチディスクへデータの出 力を行う	7 6 D 5
DSKVfy	DMA を使用しないで 3 インチ or 5 インチディスクのベリファイ を行う	7 6 E 0
MOTOF 8	3 インチ or 5 インチ or 8 インチディスクドライブのモーター OFF	7 7 9 2
MOTOFF	3 インチ or 5 インチディスクのモーター OFF	7 7 9 7
HDINIT	ハードディスクのイニシャライズ	7 8 D 9
HD0FFS	BASIC の HDOFF と同じ動作	7 8 E 2
HDDMAS	ハードディスクのリード／ライト	F 9 2 9
DSKWKS	コマンドを送出後、ディスク 1 セクターリード／ライト	F 9 6 E

その他

IPLBOT	IPL のコールドスタート	0 0 0 0
WORKBS	BIOS ワークエリア(F800H -¼ FEFFH)のイニシャライズ	1 0 6 C
BIOSIN	BIOS ワークエリアと各 I / O のイニシャライズ	1 0 8 5
BIOSRS	各 I / O のイニシャライズ	1 0 9 9
FNMTCH	HL で示されたアドレスから格納されているファイルコントロール ブロック(FCB)の内容と DIRIMG の内容が一致するかのチェック	3 A 0 3
SETDIR	HL で示されたアドレスから格納されている FCB の内容の DIR- IMG への転送	3 A 4 3
MONOP	モニターサブルーチン	3 3 C 5
JPBCNE	BC レジスタの示す ROM 内ルーチンへのジャンプ	7 D 6 C
INTSUB	割り込み処理ルーチンへのジャンプ	F 8 4 7
MEMEMM	MEM: EMM: のリードライト	F 9 A 4
HLDECK	HL, DE の示すアドレスからの値の C バイト比較	F A 2 5
\$ 0 0 F 5	IPL が正常に BOOT できなかった場合のエラー処理	0 0 F 5
DIOERR	デバイス I / O エラー(コード 56)を返す	1 0 0 0
TWRTPR	書き込み禁止エラー(コード 72)を返す	1 0 0 3
DEVUNA	デバイスオフラインエラー(コード 73)を返す	1 0 0 6

IFCALL	イリーガルファンクションコールエラーの(コード5)を返す	I 0 0 E
OVERFL	オーバーフローエラー(コード6)を返す	I 0 1 1
DVBYZR	ディビジョンバイゼロエラー(コード11)を返す	I 0 1 4
TYPEMS	タイプミスマッチエラー(コード13)を返す	I 0 1 7
TOOCMP	トゥーコンプレックスエラー(コード16)を返す	I 0 1 A
BADSMD	バッドスクリーンモードエラー(コード25)を返す	I 0 1 D
FOVER	フォーマットオーバーエラー(コード36)を返す	I 0 2 0
BADFDC	バッドファイルディスクリプターエラー(コード65)を返す	I 0 2 3
BADREC	バッドレコードエラー(コード66)を返す	I 0 2 6
PNORDY	プリンターオフラインエラー(コード73)を返す	I 0 2 9
BADFMD	バッドファイルモードエラー(コード30)を返す	I 0 2 E
BADPAS	パスワード無しエラー(コード67)を返す	I 0 3 1
BIOSER	BIOS でエラーが発生したときの処理を行う	F 8 3 C

B-1-2 アドレス順

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
IPLBOT	0000				IPLのコールドスタート。
\$00F5	00F5				IPLが正常に起動できなかった場合のエラー処理。(注1)
\$021A	021A	HL : メモリーアドレス DE : レコード番号 A : レコード長 FF87H : ドライブ番号 FF8AH : エラージャンプアドレス	HL : 次のメモリーアドレス	AF, AF', BC, DE	IPLからの2D 3インチまたは5インチディスクのリード。
\$038A	038A	FF85Hにキーデータ	A : キーコード	フラグ	IPLのキー入力。
\$03CB	03CB	DE : メッセージスタートアドレス FF86H : 色 FF80H : X座標 FF81H : Y座標	DE : エンドマークのアドレス	AF	IPLのメッセージ出力。(注2)
\$03D9	03D9	A : 出力コード FF86H : 色 FF80H : X座標 FF81H : Y座標	A : 出力コード FF80H : 次のX座標 FF81H : 次のY座標		IPLの一文字表示。
DAYBUF	0E88				曜日のメッセージ用データのテーブル。
DIOERR	1000		A : エラーコード		デバイスI/Oエラーのコード56を返す。
TWRTPR	1003		A : エラーコード		書き込み禁止エラーのコード72を返す。
DEVUNA	1006		A : エラーコード		デバイスオフラインエラーのコード73を返す。
IFCALL	100E		A : エラーコード		イリーガルファンクションコールエラーのコード5を返す。
OVERFL	1011		A : エラーコード		オーバーフローエラーのコード6を返す。
DVBYZR	1014		A : エラーコード		ディビジョンバイゼロエラーのコード11を返す。
TYPEMS	1017		A : エラーコード		タイプミスマッチエラーのコード13を返す。
TOOCMP	101A		A : エラーコード		トゥーコンプレックスエラーのコード16を返す。
BADSMD	101D		A : エラーコード		バッドスクリーンモードエラーのコード25を返す。
FOVER	1020		A : エラーコード		フォーマットオーバーエラーのコード36を返す。
BADFDC	1023		A : エラーコード		バッドファイルディスクリプタエラーのコード65を返す。
BADREC	1026		A : エラーコード		バッドレコードエラーのコード66を返す。
PNORDY	1029		A : エラーコード		プリンタオフラインエラーのコード73を返す。
BADFMD	102E		A : エラーコード		バッドファイルモードエラーのコード30を返す。
BADPAS	1031		A : エラーコード		パスワード無しエラーのコード67を返す。
WORKBS	106C	SP (F800H - FEFH以外)		HL, DE, BC	BIOSワークエリア(F800H - FEFH)のイニシャライズを行います。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
BIOSIN	1085	SP (F800H - FEFH 以外)		HL, DE, BC, AF HL', DE', BC', AF'	BIOS ワークエリアと各 I/O の イニシャライズを行います。(注 3)
BIOSRS	1099	COLORF, WK1FD0 CLSCHR SCRMOD		HL, DE, BC, AF HL', DE', BC', AF'	各 I/O のイニシャライズを行 います。(注4)
PSGINT	10B4			HL, BC, AF	PSG データのイニシャライズ を行います。(注5)
KVECIN	10D0			HL, AF, I	SUB CPU(80C49)に対し、キー 入力のインターラプトモードを セットします。(注6)
KVEC00	10D6	L:00H		AF	SUB CPU(80C49)に対し、キー 入力のノンインターラプトモード をセットします。
SCRNSB	10DF	A:スクリーンモード WIDTH0		HL, DE, BC, AF HL', DE', BC', AF'	スクリーンモードの設定を行 います。(注7)
CR400S	11D8			HL, DE, BC, AF	CRTC を400ラインに設定しま す。(注8)
ROMASK	11E7	COLORF, CLSCHR SCRMOD, WK1FD0		HL, DE, BC, AF HL', DE', BC', AF'	BASIC の ASK コマンドの処 理を行います。
WIDTH80	1220	COLORF, CLSCHR SCRMOD, WK1FD0 GRAYMX, CURYMX		HL, DE, BC, AF HL', DE', BC', AF'	WIDTH 80 の設定を行います。 (注9)
WIDTH40	1227	COLORF, CLSCHR SCRMOD, WK1FD0 GRAYMX, CURYMX		HL, DE, BC, AF HL', DE', BC', AF'	WIDTH 40 の設定を行います。 (注10)
CTRLD2	12B9	WIDTH0, CURYMX WK1FD0, SCRNM3		AF	コンソールをイニシャライズし て SCREEN 0, 0 を行います。

注1: "Make your device ready"と表示して新たな Key 入力を待ちます。

注2: メッセージスタートアドレスからアスキーコードで文字を格納します。エンドマークは 00H です。

注3: WORKBS と BIOSRS を合わせたものです。106CH, 1099H 参照

注4: 各 I/O とは、80C49, スクリーン, パレット, キーベクタ, SIO, CTC, PSG のことです。

注5: BASIC における SOUND 7, &H38: SOUND 8, 0: SOUND 9, 0: SOUND 10, 0 と同じ処理を行います。

注6: レジスタ I には 0F8H(キー割り込み処理のベクタ上位アドレス)がセットされます。

注7: BASIC における WIDTH, ?, ?, ? と同様の処理

以下に A レジスタのとり値におけるスクリーンモードを示します。

A レジスタ	スクリーンモード	テキスト行数	グラフィック
*0H	WIDTH, 25, 0	25 行	200ライン
*1H	, 12, 0	12 行	192ライン
*2H	, 20	20 行	なし
*3H	, 10	10 行	なし
*4H	, 25, 1	25 行	400ライン
*5H	, 12, 1	12 行	384ライン
0 *H	WIDTH, , , 0		200/400 セレクトスイッチ
1 *H	, , , 1		200ライン ディスプレイ
2 *H	, , , 2		400ライン ディスプレイ

注8: CRTC のレジスタ R0, R3, R4~R9 をセットします。

注9: BASIC の WIDTH80 と同じ処理をします。

注10: BASIC の WIDTH40 と同じ処理をします。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
SCRNOT	1 2 D B	A:ディスプレイモード WK1FD0 SCRNM3		AF	スクリーンのディスプレイモードの設定を行います。(注1)
SCRNIN	1 3 0 7	A:アクセスモード WK1FD0 SCRNM3		AF	スクリーンのアクセスモードの設定を行います。(注1)
PALETI	1 3 4 C	TPRIOF		D, BC, AF	パレットのイニシャライズとプライオリティの設定を行います。
STPRIO	1 3 5 9	BPRIOF RPRIOF GPRIOF TPRIOF		D, BC, AF	パレットとプライオリティの設定を行います。
PALETF	1 3 6 C			BC, AF	パレットとプライオリティを全て0にします。
STCLST	1 3 7 7	COLORF CLSCHR		HL, D, BC, AF	テキストV-RAMをクリアします。
STCLSG	1 3 9 A	WK1FD0 SCRMOD		BC, AF	グラフィックV-RAMをクリアします。
S49RES	1 3 E 5			BC, DE, AF	SUB CPU(80C49)のコミュニケーションバッファをクリアします。
IN49SB	1 4 0 8		A:入力データ	フラグ	SUB CPU(80C49)よりデータを1バイト入力します。
OT49SB	1 4 1 3	A:出力データ	A:出力データ	フラグ	SUB CPU(80C49)へデータを1バイト出力します。
COMOUT	1 4 3 2	A:コマンドデータ		AF	SUB CPU(80C49)へコマンドを出力します。
TAK49S	1 4 3 B	A:コマンド DE:データバッファ		DE, B, AF	SUB CPU(80C49)とZ-80Aとの間でデータの受け渡しをします。(注2)
PALSET	1 4 8 0	D:パレットレコード E:カラーコード BPRIOF GPRIOF RPRIOF		DE, AF	パレットのカラーを設定します。(注3)
INTCRT	1 4 B F	WIDTH0 GRAXMX WK1FD0 GRAYMX CURYMX		I, IX, IY 以外すべて	スクリーンのイニシャライズを行います。(注4)
CRTCR1	1 6 C 5	CSIZEF CURX CURY		AF	CSIZEサポートのCR+LFの出力を行います。(注5)
CRTACC	1 6 D 3	A:表示文字コード CSIZEF CURX CURY COLORF			CSIZEサポートの一文字出力を行います。(注6)
DEPRT	1 7 5 4	DE:文字列のスタートアドレス CURX CURY COLORF	DE:文字列のエンド アドレス+1		画面へ文字列を出力します。
CR2	1 7 7 0				現在のカーソルのX座標が0以外の時、CR1へ飛び改行を行います。
CR1	1 7 7 8	CURX CURY		AF	改行を行います。
TABPRT	1 7 8 0	CURX CURY COLORF		AF	HTAB PRINTの処理をします。(注7)

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
SPPRT	178F	CURX CURY COLORF		A	画面にスペースを表示します。
ACCPRT	1791	A：表示文字コード CURX CURY COLORF	A：表示文字コード		画面に一文字表示します。 (注8)
ACCDIS	179D	A：表示文字コード CURX CURY COLORF	A：表示文字コード		画面に一文字表示します。 (注9)
TBCALC	18B1	H：Y座標 SCRNTC	HL：コネクトフラグ アドレス E：Y座標	フラグ D：00H	テキストV-RAM上の任意の 行のコネクトフラグのアドレス を出力します。(注10)

注1：以下にアクセスモードの値を示します。

Aレジスタ	テキストページ	グラフィックページ
00H	0	0
01H	1	1 (WIDTH 40 のみ)
02H	0	2 (グラフィック 200, 192ラインのみ)
03H	1	3 (WIDTH 40/200, 192ラインのみ)

注2

コマン ド	内 容	後続データ バイト数
D0H~D7H	タイマ0~7をセットする	6
D8H~DFH	タイマ0~7の内容をリードする	6
E3H	ゲームキーデータリード	3
E4H	KEY 割り込みベクタ値をセット (キーベクタアドレスの LOWBYTE を返す) (ただし、0の場合は、割り込み禁止モードになる)	1
E5H	タイマオールクリア	0
E6H	キーバッファリード (80C49のキーバッファの内容を Z-80A へ送る)	2
E7H	TV 送信コードセット	1
E8H	TV 送信コードリード (TV へ最後に送られたコードを Z-80A へ返す)	1
E9H	データレコーダコントロールコマンドのセット	1
EAH	データレコーダの動作状態の読み出し	1
EBH	カセットセンサリード	1
ECH	日付のセット	3
EDH	日付のリード	3
EEH	時刻のセット	3
EFH	時刻のリード	3

*タイマ0は BASIC の ON TIMES\$ GOSUB など使われるシステム用タイマです。

注3：BASICにおける PALET(パレットコード)、(カラーコード)と同じ働きをします。

注4：BASICにおける INIT^CRT^と同じ働きをします。

注5：CSIZEFのビット0が0：1回改行 1：2回改行

注6：BASICにおける PRINT #0と同じ働きをします。

注7：PRINT A, B の", "の処理のようにカーソルのX座標を10の倍数の位置へ持っていきます。

注8：表示文字コードがコントロールコード(00H~1FH)の場合、CTRLJB(18E1H)へジャンプします。

注9：ACCPRT とのちがいは、コントロールコード(00H~1FH)の場合でもそのまま文字として表示することです。

注10：コネクトフラグはカーソルのある行が先頭行なら0、継続行なら1を表示します。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
ADRCA2	18BC	L:X座標 H:Y座標	HL:テキストI/O アドレス 3000H~37FFH	AF, BC	現在のカーソル位置のテキスト V-RAM上でのアドレスを計 算します。
CURADR	18C5	CURX CURY	HL:オフセットアド レス 0000H~07FFH	AF, BC	ADRCA2のサブルーチンです。
CTRLJB	18E1	A:コントロールコー ド		AF, BC DE, HL	コントロールコード(00H ~1FH)の出力の処理をします。
BEEP	1B41			AF, BC DE, HL	BEEP音を出します(=CTRL -G)。
STRIGS	1D89	A:モード (注3参照)	A:ONの時20H OFFならそれ以 外	AF, BC, HL	ジョイスティックのトリガまた はスペースキーの状態を調べま す。(注1)
STICKS	1D92	A:モード STRIGS参照	A:1~9のアスキー コード	AF, BC, HL	ジョイスティックの方向, また はテンキーの値を調べます。
BINPUT	1DC2	DE:データ格納アド レス	DE:データ格納アド レス CY	AF	BASICのINPUT文と同じ処 理をします。(注2)
INPUTF	1DE4	DE:データ格納アド レス	DE:データ格納アド レス CY	AF	BASICのLINPUT文と同じ 処理をします。(注3)
BCUYST	1F16	H:CURY	D E=CURY HL=フラグアドレス	AF, DE, HL	HにY座標を入力し, その行の 始まっているY座標をDに返し ます。
ECUYST	1F25	H:CURY	D E=CURY HL=フラグアドレス	AF, DE, HL	HにY座標を入力し, その行の 終わり+1のY座標をDに返し ます。
SCRGET	1F8F	A:読み込む文字数 E:X座標 D:Y座標 HL:データ格納アド レス		AF, AF', BC, DE, HL	BASICのSCRN\$と同じ働き をします。
INKEYS	1FF0	A(注1)	A:文字コード	フラグ	キーボードから一文字入力しま す。(注4)
BRKCKS	20D5		ZF	AF	SHIFT+BREAKまたはCT RL-Cが押されているかどうか を調べます。(注5)
KEYSNS	20EB		ZF	AF	入力されたデータが有効かどう かを調べます。(注6)
KEYSN1	20F5		ZF	AF, BC, DE, HL	ROM用のKEYSNSサブルー チン。(注6)
X1HPDS	274E	D:INKEY\$(2)7DH KEYDAT+1 X1HELP WIDTH0 X1MODE		AF, BC, DE, HL	XFERモードの表示をします。
FKYDSI	2A1B			AF, BC, DE, HL	ファンクションキーを表示しま す。
FKYDSS	2A22	D=INKEY\$(2)		AF, BC, DE, HL	ファンクションキーモードを表 示します。
EDLNDS	2A6B	X1MODE FKYDSF WIDTH0		AF, BC, DE, HL	XFER/ファンクションモード の表示をします。
SFTKTN	2F07	DE:シフトJISコー ド (HEX)	DE:区点コード (BCD)	AF	シフトJISコード→区点コード の変換を行います。(注7)
KTNSFT	2F2C	DE:区点コード (BCD)	DE:シフトJISコー ド (HEX)	AF	区点コード→シフトJISコード の変換を行います。
JISSFT	2F52	DE:JIS漢字コード (HEX)	DE:シフトJISコー ド (HEX)	AF	JIS漢字コード→シフトJISコ ードの変換を行います。(注8)

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
SFTJIS	2F81	DE: シフト JIS コード (HEX)	DE: JIS 漢字コード (HEX)	AF	シフト JIS コード→ JIS 漢字コードの変換を行います。
JISVRM	2FB6	DE: JIS 漢字コード (HEX)	A E D CY	フラグ	JIS 漢字コード→ V-RAM データの変換を行います。(注9)
VRMJIS	3037	A E D	DE: JIS 漢字コード (HEX)	フラグ	V-RAM データ→ JIS 漢字コードの変換を行います。(注10)
SFTCHK	3099	A: チェックする 1バイトデータ	CY	フラグ	シフト JIS コードの上位1バイトかどうかを調べます。(注11)
KANDAK	30A3	DE: JIS 漢字コード (HEX)	DE: JIS 漢字コード (HEX) CY	AF	JIS 漢字コードを濁点付きのコードに変換します。(注12)
KANHAN	30F2	DE: JIS 漢字コード (HEX)	DE: JIS 漢字コード CY	AF	JIS 漢字コードを半濁点付きのコードに変換します。(注13)
ASCKAN	3119	A: アスキーコード	DE: JIS 漢字コード	AF	アスキーコードを JIS 漢字コードに変換します。(注14)
KANASC	31A6	DE: JIS 漢字コード	DE CY	AF, DE	JIS 漢字コードをアスキーコードに変換します。(注15)
CGSET	32AD	DE: アスキーコード 外字 JIS コード HL: データバッファ	CY	AF, BC DE, HL	PCG の定義をします。(注16)
CGREAD	330D	DE: アスキーコード JIS 漢字コード HL: データバッファ アドレス	CY DE HL	AF, BC, DE	CG のパターンデータを読み込みます。(注17)
MONOP	33C5	HL: DUMP ADDRESS ????		AF, BC, DE, HL, AF', BC', DE', IX, IY	モニターサブルーチンです。

- 注1: Aにセットする値は、スペースキーを調べる時0、ジョイスティック1を調べる時1、ジョイスティック2を調べるとき2です。STICKSのときも同じです。
- 注2: CY=0のときCRがCTRL-Jによる正常な入力、CY=1でA=3のときSHIFT+BREAKかCTRL-Cによって入力が終了、CY=1でA=4のときCTRL-Dによる入力の終了です。
- 注3: BINPUTを参照してください。
- 注4: A=FFHならINKEY\$, A=00HならINKEY\$(0), A=01HならINKEY\$(1), A=02HならINKEY\$(2)となります。
- 注5: ZF=0のとき押されていない、ZF=1のとき押されていることを示します。
- 注6: ZF=0のとき入力されたデータは有効となります。
- 注7: シフト JIS コード 8140H~EFFCH, 区点コード0101H~9494H
- 注8: JIS 漢字コード 2121H~7E7EH
- 注9: Aには、アトリビュート V-RAM データ(2000H~27FFH)bit5のみ
Eにはテキスト V-RAM データ(3000H~37FFH)
Dには漢字テキスト V-RAM データ(3800H~3FFFH)
が出力されます。但し、CY=1となっている時、入力されたコードは JIS 漢字コードではなかったことを示します。
- 注10: JISVRMを参照して下さい。
- 注11: Aレジスタに入力された1バイトのデータがシフト JIS コードの上位1バイトであればCY=0となります。
- 注12: 例えば、「か」という字を引数に、このルーチンを読んだ場合、DEには「が」の漢字コードが返されます。濁点をつけることのできない字、例えば「あ」を引数にした場合にはCY=1とした上でDEには「^a」の文字コード212BHが返されます。
- 注13: KANDAKと同じ処理を半濁点について行います。
- 注14: 半角アスキー文字の文字コードを JIS 全角文字コードに変換します。アスキーコードの範囲外のときにはDEには「※」のコード2228Hが返されます。
- 注15: ASCKANの逆の動作を行います。例えば、全角の「A」を引数にこのルーチンを読んだ場合、Dには半角アスキー文字「A」のアスキーコード**Hが返されます。引数が、範囲外(例えば「垂」)の場合CY=1としてDEにはそのままコードが返されます。
- 注16: データバッファはアスキーコードのとき(B8+R8+G8バイト)×1
外字 JIS コードのとき(B8+R8+G8バイト)×4です。
CY=1の場合はDEレジスタが範囲外で実行できなかった場合です。
- 注17: 正常に実行できた場合、CY=0となりHLに次のデータバッファアドレスがセットされますが、DEレジスタが範囲外で実行できなかった時はCY=1となります。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
ACSET	3 6 1 6	DE: データアドレス	DE: 次のデータアドレス A	フラグ	(注1)
CR1PRP	3 7 A B				FILOUT=0 → CR1 FILOUT=1 → CR1LPL へと 処理を渡します。
CR1LPL	3 7 B 2			AF	プリンターへCRとLFのロード を出力してからLPOSをク リアします。
CR1LPT	3 8 0 F			AF	プリンターへCRとLFのコード を出力します。
ACCPRP	3 8 3 1	A: 出力する文字コード	A: 出力する文字コード	AF'	FILOUT=0 → ACCPRT FILOUT=1 → ACCLPL へと 処理を渡します。
ACCLPL	3 8 3 9	A: 出力する文字コード	A: 出力する文字コード	フラグ	プリンターへの1文字出力を行 います。(注2)
HLLPRT	3 9 2 7	HL: データテーブル 先頭アドレス	HL: 最終データ格納 アドレス	フラグ	HLレジスタの示すアドレスから 始まるデータテーブルのデー タをプリンターに出力します。 (注3)
ACCLPT	3 9 8 3	A: 出力する文字コード	A: 出力する文字コード	フラグ	プリンターへ1文字を出力しま す。
LPTSNS	3 9 A 1	PRTDLY	CY: 1 TIME OUT CY: 0 READY	AF, BC D, HL	プリンターの状態を調べます。
TABPRP	3 9 B A				FILOUT=0 → TABPRT FILOUT=1 → TABLPL へと 処理を渡します。
TABLPL	3 9 C 1			AF	プリンターへの水平タブの出力 を行います。(注4)
FMPRHL	3 9 D 6	DE HL		AF, DE	CR1を行ったあと「Found」フ ァイルネーム、拡張子」か 「Writing」ファイルネーム、拡 張子」の表示を行います。 (注5)
FNMTCH	3 A 0 3	HL: ロードされた FCBの格納アド レス	ZF=1 すべて一致 ZF=0 一致しない。	AF, B	HLで示されたアドレスから格 納されているファイルコント ロールブロック(FCB)の内容と DIRIMGの内容が一致するか 調べます。
SETDIR	3 A 4 3	HL: FCBの格納アド レス		AF, B, DE, HL	HLで示されたアドレスから格 納されているFCBの内容を DIRIMGへ転送します。
WRTMES	3 A D 2				「Writing」のメッセージデータ テーブル。
FINMES	3 A D A				「Found」のメッセージデータテ ーブル。
SKPMES	3 A E 2				「Skip」のメッセージデータテ ーブル。
SUB	3 A F 8	HL: データ1格納アド レス DE: データ2格納アド レス PRCSN: データタイプ (2, 5, 8)	HL: 結果のデータ1 格納アドレス DE: データ2格納アド レス PRCSN: 結果のタイプ	AF, BC, AF', BC', DE', HL'	[HL]=[HL]-[DE]を行います。 (注6)

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
ADD	3 A F B	HL: データ 1 格納アドレス DE: データ 2 格納アドレス PRCSN: データタイプ (2, 5, 8)	HL: 結果のデータ 1 格納アドレス DE: データ 2 格納アドレス PRCSN: 結果のタイプ	AF, BC, AF', BC', DE', HL'	[HL]=[HL]+[DE]を行います。
CMP	3 D B A	HL: データ 1 格納アドレス DE: データ 2 格納アドレス PRCSN: データタイプ (2, 5, 8)	HL: データ 1 格納アドレス DE: データ 2 格納アドレス CY: 1 なら違う ZF: 1 なら同じ	AF, B	[HL], [DE]の比較を行います。
MUL	3 E 0 1	HL: データ 1 格納アドレス DE: データ 2 格納アドレス PRCSN: データタイプ (2, 5, 8)	HL: データ 1 格納アドレス DE: データ 2 格納アドレス PRCSN: 結果のタイプ	AF, BC, AF', BC', DE', HL'	[HL]=[HL]×[DE]を行います。
DIV	4 0 3 E	HL: データ 1 格納アドレス DE: データ 2 格納アドレス PRCSN: データタイプ (2, 5, 8)	HL: データ 1 格納アドレス DE: データ 2 格納アドレス PRCSN: 結果のタイプ	AF, BC, AF', BC', DE', HL'	[HL]=[HL]÷[DE]を行います。
INTDVS	4 0 E 3	DE: データ 1 HL: データ 2	DE: 商 HL: 余り	AF, BC	符号付整数の除算 DE÷HL=DE…HL
INTDVN	4 1 1 D	DE: データ 1 HL: データ 2	DE: 商 HL: 余り	AF, BC	符号無し整数の除算 DE÷HL=DE…HL
INTDVV	4 1 2 2	HL: データ 1 の上位 DE: データ 1 の下位 BC: データ 2	DE: 商 HL: 余り	AF, BC	符号無し整数の除算 HLDE÷BC=DE…HL HL<BC
CVFLAS	4 3 5 3	DE: アスキー文字列先頭アドレス HL: 結果格納アドレス	DE: アスキー文字列の最後のアドレス+1 HL: 結果格納アドレス PRCSN: 結果のタイプ	AF, BC, AF', BC', DE', HL', IX, IY	アスキー文字列を浮動小数点型データに変換します。(注7)

注1: DEで指定したアドレスからデータを読み込み、そのデータをアスキーコードとしてデータの表す文字を得ます。連続した2つのデータを文字に直してそれが16進数を示しているなら数値化してAに入れDEには、そこまで読み込まれた最後のデータの次のデータの入っているアドレスがセットされます。またスペース(=20H)はとばされます。またそれ以外であればDEが示すアドレスの内容がAに入ります。
このルーチンはDEが示すアドレスから入っている文字列によって動作が違います。

注2: LPOS=LPOS+1となります。

注3: HLで示されたアドレスすなわちデータテーブルの先頭アドレスには出力長が入っています。

注4: BASICにおけるLPRINT A, B, Cに関する処理です。

注5: DEには「Loading」, 「Writing」のメッセージのある先頭アドレスを設定します。HLにはファイルネームのある先頭アドレスをセットします(ファイルネーム13文字、拡張子3文字)。

注6: PRCSNには、2→整数型、5→単精度型、8→倍精度型がはいります。結果は、出力されたHLの示すアドレスに格納されます。DEは変化しません。PRCSNも変化しません。[HL] [DE]などは、HLを示すアドレスの内容、DEを示すアドレスの内容です。以下の計算サブルーチンでもこれに準じます。

注7: 2EH=「,」か30H~39Hの10進数を表すアスキーコードで文字列は構成されている必要があります。それ以外のコードが出たところでデータ終了です。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
ANDBOH	4 4 9 4	DE	DE: 文字列終了アドレス+1 HL: 結果 CY=1 なら オーバーフロー	AF, BC	アスキー文字列を整数型データに変換します。(注1)
CHCKHX	4 4 E 7	A: アスキーコード	A: 00H~0FH CY=1 なら範囲内 CY=0 なら範囲外	AF	Aレジスタの値が16進数を表すアスキーコードかどうかをチェックします。(注2)
CVHLAS	4 4 F 5	DE: アスキー文字列先頭アドレス A: タイプ	DE: 終了アドレス+1 HL: 結果 CY=1 なら オーバーフロー	AF, BC	Aレジスタにタイプを入力し、数値を表す文字列を数値に変換します。(注3)
HEXCUL	4 4 F A	DE: アスキー文字列先頭アドレス	DE: 終了アドレス+1 HL: 結果 CY=1 なら オーバーフロー	AF, BC	10進数を表すアスキー文字列を数値に変換します。
TOGLE	4 5 2 6	HL: データの先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: 結果の先頭アドレス PRCSN: 結果のタイプ	AF	[HL] = -[HL]を行います。
MULTEN	4 5 6 5	HL: 浮動小数点型データの先頭アドレス PRCSN: データタイプ (5, 8)	HL: 結果の先頭アドレス PRCSN: 結果のタイプ	AF', BC', DE', HL', IX, IY	[HL] = [HL] × 10
DIVTEN	4 5 7 2	HL: 浮動小数点型データの先頭アドレス PRCSN: データタイプ (5, 8)	HL: 結果の先頭アドレス PRCSN: 結果のタイプ	AF', BC', DE', HL'	[HL] = [HL] ÷ 10
MULDEC	4 5 7 F	HL: 加算データ1アドレス A: 加算データ2 PRCSN: データタイプ (5, 8)	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, AF', BC', DE', HL'	[HL] = [HL] + A
FLTHEX	4 5 A 6	DE: 整数型データ HL: データバッファアドレス(8バイト)	HL: 結果のデータの先頭アドレス	AF, B, DE	整数型データ→浮動小数点型データの変換を行います。
CVNMFL	4 5 D 2	HL: データ先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: データ先頭アドレス DE: アスキー文字列先頭アドレス	AF, BC, DE, AF', BC', DE', HL', IX, IY	浮動小数点型データ→符号付アスキー文字列への変換を行います。
CVASFL	4 5 F 3	HL: データ先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: データ先頭アドレス DE: アスキー文字列先頭アドレス	AF, BC, DE, AF', BC', DE', HL', IX, IY	浮動小数点型データ→符号無しアスキー文字列への変換を行います。
CVASIN	4 6 A E	HL: 整数型データ先頭アドレス	HL: データ先頭アドレス DE: アスキー文字列先頭アドレス	AF, DE	整数型データ→符号無しアスキー文字列への変換を行います。(注4)
CVASII	4 6 B 8	HL: 整数型データ	HL: 整数型データ DE: アスキー文字列先頭アドレス	AF, DE	HLに入っている整数型データ→アスキー文字列への変換を行います。(注4)

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
CVASSN	4 6 CA	HL: 整数型データ先頭アドレス	HL: データ先頭アドレス DE: アスキー文字列先頭アドレス	AF, DE	整数型データ→符号付アスキー文字列への変換を行います。(注4)
ASC FIV	4 6 E 7	HL: 整数型データ	HL: 整数型データ DE: アスキー文字列先頭アドレス	AF, B, DE	整数型データ→符号無しアスキー文字列への変換を行います。(注4)
HEXHL0	4 6 F 1	HL: 整数型データ	HL: 整数型データ DE: アスキー文字列先頭アドレス	AF, B, DE	HLに入っている整数型データ→16進数を表すアスキー文字列への変換を行います。(注5)
BINHL0	4 6 FB	HL: 整数型データ	HL: 整数型データ DE: アスキー文字列先頭アドレス	AF, B, DE	HLに入っている整数型データ→2進数を表すアスキー文字列への変換を行います。(注5)
OCTHL0	4 7 0 5	HL: 整数型データ	HL: 整数型データ DE: アスキー文字列先頭アドレス	AF, B, DE	HLに入っている整数型データ→8進数を表すアスキー文字列への変換を行います。(注5)
ASCHL	4 7 1 5	HL: 整数型データ DE: アスキー文字列先頭アドレス	HL: 整数型データ DE: アスキー文字列先頭アドレス	AF, DE	HLに入っている整数型データ→10進数を表すアスキー文字列への変換を行います。(注6)
BINHL	4 7 4 7	HL: 整数型データ DE: アスキー文字列先頭アドレス	HL: 整数型データ DE: アスキー文字列先頭アドレス	AF, DE	HLに入っている整数型データ→2進数を表すアスキー文字列への変換を行います。(注7)
OCTHL	4 7 5 6	HL: 整数型データ DE: アスキー文字列先頭アドレス	HL: 整数型データ DE: アスキー文字列先頭アドレス	AF, DE	HLに入っている整数型データ→8進数を表すアスキー文字列への変換を行います。(注8)
KT NHL	4 7 6 F	HL: 整数型データ (シフトJISコード)	HL: 整数型データ (区点コード) DE: アスキー文字列先頭アドレス	AF, DE	HLに入っているシフトJISコード→区点コードを表すアスキー文字列への変換を行います。
JISHL	4 7 7 5	HL: 整数型データ (シフトJISコード)	HL: 整数型データ (JIS漢字コード) DE: アスキー文字列先頭アドレス	AF, DE	HLに入っているシフトJISコード→JIS漢字コードを表すアスキー文字列への変換を行います。
HEXHLB	4 7 7 9	DE: 整数型データ	DE: アスキー文字列先頭アドレス	AF, DE, HL	DEに入っている整数型データ→16進数を表す符号無しアスキー文字列への変換を行います。(注9)
HEXHL	4 7 7 D	HL: 整数型データ DE: アスキー文字列格納先頭アドレス	HL: 整数型データ DE: アスキー文字列格納先頭アドレス	AF HL	HLに入っている整数型データ→16進数を表すアスキー文字列に変換してDEで指定したアドレスに格納します。(注9)

注1: DEには「^&」アンパサンド=アスキーコード26H」の次のアスキーコード文字があるアドレスを入力します。&の次の文字がB(42H)なら2進数, O(4FH)なら8進数, H(48H)なら16進数, J(4AH)ならJIS漢字コード, K(4BH)ならJIS区点コードを表します。

注2: Aレジスタが30H~39H, 41H~46H, 61H~66Hであるかを調べてそうである場合は, 30H~39Hは00H~09H, 41H~46Hは0AH~0FH, 61H~66Hも0AH~0FHの値がAレジスタに設定されます。

注3: A=44H →10進数 A=4FH →8進数 A=4AH JIS漢字コード
A=42H →2進数 A=48H →16進数 A=4BH JIS区点コード

注4: アスキー文字列のエンドコードは00Hです。

注5: アスキー文字列のエンドコードは, 00Hです。また, アスキー文字列の先頭が30H(^0")であるとき出力されたDEはその次のアドレスを示しています。

注6: OCTHL0と違い, アスキー文字列の先頭が30Hでも出力されたDEは, 30Hがあるアドレスを示します。従って, 常に5バイトの文字列に変換されます。アスキー文字列エンドコードは00Hです。

注7: BINHL0と違い, アスキー文字列の先頭が30Hでも, 出力されたDEは, 30Hがあるアドレスを示します。従って常に16バイトの文字列に変換されます。アスキー文字列エンドコードは00Hです。

注8: OCTHL0と違い, アスキー文字列の先頭が30Hでも出力されたDEは, 30Hがあるアドレスを示します。従って, 常に6バイトの文字列に変換されます。アスキー文字列エンドコードは00Hです。

注9: アスキー文字列の先頭が30Hでも出力されたDEは30Hがあるアドレスを示します。従って常に4バイトの文字列に変換されます。アスキー文字列エンドコードは00Hです。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
HEXA	4 7 8 A	A: HEX データ DE: アスキー文字列 格納先頭アドレス	DE: アスキー文字列 格納先頭アドレス+1	AF	Aに入っている HEX データ→ 16進数を表すアスキー文字列に 変換して DE で指定したアドレ スから格納します。
USNGCV	4 9 0 8	HL D E A PRCSN	DE: アスキー文字列 先頭アドレス	AF, BC, AF', BC', DE', HL', IX, IY	書式指定による浮動小数点型デ ータ→アスキー文字列への変換 を行います。(注1)
HEXFLT	4 A 6 E	HL: データ先頭アド レス	HL: 整数型データ 但しオーバーフローの 時は CY=1 で HL=0 となります。	AF	HL で示されたアドレスからの 浮 動 小 数 点 型 データが -32768<[HL]<65535であ れば、整数型に変換した後 HL レ ジスタに代入します。
HLFLT	4 A 7 B	HL: データ先頭アド レス	HL: 整数型データ	AF	HL で示されたアドレスからの 浮 動 小 数 点 型 データが -32768<[HL]<65535であ れば、整数型に変換した後 HL レ ジスタに代入します。(注2)
HLFLT0	4 A 8 2	HL: データ先頭アド レス	HL: 整数型データ	AF	HL で示されたアドレスからの 浮 動 小 数 点 型 データが -32768<[HL]<65535であ れば、整数型に変換した後 HL レ ジスタに代入します。(注3)
POWERS	4 A D 9	HL: データ 1 先頭アド レス DE: データ 2 先頭アド レス PRCSN: データ 1 のタイプ POWERF: データ 2 のタイプ MEMMAX: データ格 納上限アド レス	HL: 結果格納アドレ ス PRCSN: 結果のタ イプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	[HL]=[HL] ^[DE] を行います。
ABS	4 B 8 2	HL: データ先頭アド レス PRCSN: データタ イプ	HL: 結果格納アドレ ス PRCSN: 結果のタ イプ	AF	[HL]=ABS[HL]を行います。
INTOPR	4 B 8 A	HL: データ先頭アド レス PRCSN: データタ イプ	HL: 結果格納アドレ ス PRCSN: 結果のタ イプ	AF, BC, DE, AF', BC', DE', HL'	[HL]=INT[HL]を行います。
SQR	4 B A E	HL: データ先頭アド レス PRCSN: データタ イプ MEMMAX: データ格 納上限アド レス	HL: 結果格納アドレ ス PRCSN: 結果のタ イプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	[HL]=SQR[HL]を行います。
SUM	4 B C 3	HL: データ先頭アド レス PRCSN: データタ イプ MEMMAX: データ格 納上限アド レス	HL: 結果格納アドレ ス PRCSN: 結果のタ イプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	[HL]=SUM[HL]を行います。
FACG	4 B F 1	HL: データ先頭アド レス PRCSN: データタ イプ MEMMAX: データ格 納上限アド レス	HL: 結果のアドレス PRCSN: 結果のタ イプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	[HL]=FAC[HL]を計算しま す。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
ATN	4 C 3 E	HL: データ先頭アドレス PRCSN: データタイプ MEMMAX: データ格納上限アドレス	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	{HL}=ATN(HL)を計算します。
COS	4 D 0 7	HL: データ先頭アドレス PRCSN: データタイプ MEMMAX: データ格納上限アドレス	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	{HL}=COS(HL)を計算します。
SIN	4 D 2 0	HL: データ先頭アドレス PRCSN: データタイプ MEMMAX: データ格納上限アドレス	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	{HL}=SIN(HL)を計算します。
TAN	4 E 2 5	HL: データ先頭アドレス PRCSN: データタイプ MEMMAX: データ格納上限アドレス	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	{HL}=TAN(HL)を計算します。
SGN	4 E 5 C	HL: データ先頭アドレス PRCSN: データタイプ MEMMAX: データ格納上限アドレス	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, DE	{HL}=SGN(HL)を計算します。
RAD	4 E 8 4	HL: データ先頭アドレス PRCSN: データタイプ	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	{HL}=RAD(HL)を計算します。
PAI	4 E 8 D	HL: データ先頭アドレス PRCSN: データタイプ	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	{HL}=PAI(HL)を計算します。
RND	4 E 9 6	HL: データ先頭アドレス PRCSN: データタイプ	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, BC, DE, AF', BC', DE', HL'	{HL}=RND(HL)を計算します。
EXP	4 E C 5	HL: データ先頭アドレス PRCSN: データタイプ MEMMAX: データ格納上限アドレス	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	{HL}=EXP(HL)を計算します。
LOG	4 F D 8	HL: データ先頭アドレス PRCSN: データタイプ MEMMAX: データ格納上限アドレス	HL: 結果のアドレス PRCSN: 結果のタイプ	AF, BC, DE, AF', BC', DE', HL', IX, IY	{HL}=LOG(HL)を計算します。

注1: HL: データ先頭アドレス D: 整数部桁数 E: 小数部桁数 A: 指数表現をするとき1, しないとき0
PRCSN: データタイプ(5, 8)

注2: HEXFLTと違うところは, オーバーフロー時にエラー処理ルーチンへジャンプすることです。

注3: オーバーフローの時HLFLTと同じようにエラー処理ルーチンへジャンプします。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
CSNGP	5 0 B 0	HL: データ先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: 結果のアドレス PRCSN: 結果のタイプ (5, 8)	AF	[HL]=CSNG[HL]を計算します。ただし PRCSN=2 のみ実行します。(注1)
CDBL	5 1 0 2	HL: データ先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: 結果のアドレス PRCSN: 結果のタイプ(8)	AF	[HL]=CDBL[HL]を計算します。
CSNG	5 1 3 1	HL: データ先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: 結果のアドレス PRCSN: 結果のタイプ(5)		[HL]=CSNG[HL]を計算します。
CINT0	5 1 6 7	HL: データ先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: 結果のアドレス PRCSN: 結果のタイプ(2)	AF	[HL]=CINT[HL]を計算します。(注2)
CINT	5 1 7 9	HL: データ先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: 結果のアドレス PRCSN: 結果のタイプ(2)	AF	[HL]=CINT[HL]を計算します。(注3)
FIX	5 1 B E	HL: データ先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: 結果のアドレス PRCSN: 結果のタイプ (2, 5, 8)	BC', DE', HL', AF	[HL]=FIX[HL]を計算します。
FIXLT	5 1 C 4	HL: データ先頭アドレス PRCSN: データタイプ (5, 8)	HL: 結果のアドレス PRCSN: データタイプ (5, 8)	BC', DE', HL', AF	[HL]=FIX[HL]を計算します。(注4)
FRAC	5 2 5 8	HL: データ先頭アドレス PRCSN: データタイプ (2, 5, 8)	HL: 結果のアドレス PRCSN: データタイプ (2, 5, 8)	AF, AF', BC', DE', HL'	[HL]=FRAC[HL]を計算します。
CVDATS	5 2 9 6		DE: アスキー文字列先頭アドレス	AF, BC, HL	日付けを読み出しアスキー文字列で格納します。(注5)
CVDATE	5 2 9 9	HL: 日付の内部コード先頭アドレス DE: アスキー文字列格納アドレス	DE: アスキー文字列先頭アドレス	AF, BC	HL で示したアドレスから3バイトの年・月・日内部コードを読み出し DE で示すアドレスからアスキー文字列で格納します。(注6)
CVDAYS	5 2 D F		DE: アスキー文字列先頭アドレス	AF, BC, HL	曜日を読み出しアスキー文字列で格納します。(注7)
CVDAY	5 2 E 2	HL: 曜日の内部コード先頭アドレス DE: アスキー文字列格納アドレス	DE: アスキー文字列先頭アドレス	AF, BC	HL で示したアドレスから3バイトの年・月・日内部コードを読み出し DE で示すアドレスからアスキー文字列で格納します。(注8)
CVTIS\$S	5 2 F B		DE: アスキー文字列先頭アドレス	AF, BC, HL	時間を読み出しアスキー文字列で格納します。(注9)

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
CVTIME	5 3 0 0	HL: 時間の内部コード 先頭アドレス DE: アスキー文字列 先頭アドレス	DE: アスキー文字列 先頭アドレス	AF, BC	HL で示したアドレスから 3 バイトの時・分・秒内部コードを読み出し DE で示したアドレスからアスキー文字列で格納します。(注10)
CVTIMS	5 3 1 6	DE: 時間格納先頭アドレス TIMBUF TIME=0 データ	DE: 時間格納先頭アドレス PRCSN: 5 (時間タイプ)	AF, BC, HL, AF', BC', DE', HL', IX	BASIC の TIME 用の秒数を読み出し DE で示したアドレスから格納します。
DATSTS	5 3 2 B	DAYMES: 日付のアスキー文字列データ		AF, BC, DE, HL	日付を設定します。(注11)
DAYSTS	5 3 A 8	DAYMES: 曜日のアスキー文字列データ		AF, BC, DE, HL	曜日を設定します。(注12)
TISSTS	5 3 D 7	DAYMES: 時刻のアスキー文字列データ		AF, BC, DE, HL	時刻を設定します。(注13)
TISTS	5 4 1 8	DAYMES: タイムデータの内部コード PRCSN: データタイプ		AF, BC, DE, HL, AF', BC', DE', HL'	時刻を設定します。 TIME=? <input type="text"/> CR と同じです。(注14)
BOXFUL	5 5 0 7	LINEXS LINEYS LINEXE LINEYE PSMODE GCOLOR		AF, BC, DE, HL, BC', DE', HL', IX, IY	四角形を描きその内部をぬりつぶします。(注15)

注1: CSNG(5131H)との違いに注意。CSNG(5131H)参照

注2: 但し[HL]は-32768≤[HL]≤65535の範囲に限り。CINT(5179H)参照

注3: 但し[HL]は-32768≤[HL]≤32767の範囲に限り。CINT0(5167H)参照

注4: FIX(51BEH)との違いはデータのタイプが浮動小数点型のみであることです。

注5: アスキーコード 8 バイト + エンドコード (00H) 1 バイトの計 9 バイトが DE の示すアドレスから格納されています。

(例) 38 36 2F 30 33 2F 30 36 00(エンドコード)
8 6 / 0 3 / 0 6

注6: 文字列の格納状態は、CVDATS(5296H)と同じですが、その格納アドレスを DE で示すところが異なります。また内部コードは 86, 3?, 24 で月は 16 進数に直し 1 文字で曜日も 1 文字で合わせて 1 バイトです。

注7: アスキーコード 3 バイト + エンドコード (00H) 1 バイトの計 4 バイトが DE の示すアドレスから格納されています。

(例) 53 55 4E 00(エンドコード)
S U N

注8: 文字列の格納状態は CVDAYS(52DFH)と同じですがその格納アドレスを DE で示すところが異なります。また内部コードは、

86 3 / 0 24 (曜日: 日 月 火 水 木 金 土)
年 月 / 曜日 日 (コード: 0 1 2 3 4 5 6)

注9: アスキーコード 8 バイト + エンドコード (00H) 1 バイトの計 9 バイトが DE の示すアドレスから格納されています。

(例) 31 31 3A 33 32 3A 34 38 00(エンドコード)
1 1 : 3 2 : 4 8

注10: 文字列の格納状態は、CVTIS\$ (52FBH)と同じですが、その格納アドレスを DE で示すところが異なります。また内部コードは、11:32:48 なら 11 32 48 です。

注11: DAYMES は日付のアスキーコード 8 バイトが格納されているワークエリアですので、日付変更の場合は、DAYMES を設定してから実行して下さい。

注12: DATSTS と同様に DAYMES にアスキーコード 3 バイトで曜日を設定してから実行して下さい。

注13: DATSTS と同様に DAYMES にアスキーコード 8 バイトで時刻を設定してから実行して下さい。

注14: DATSTS と同様に DAYMES にセットしますが内部コードでセットするところが異なります。

注15: BASIC の LINE(X1, Y1)-(X2, Y2), モード, パレットコード, BF, TILBUF と同じ働きをします。ウィンドウからはみでた部分は、無視します。

LINEXS : 先頭 X 座標 (X1) PSMODE: モード GCOLOR : パレットコード
LINEYS : 先頭 Y 座標 (Y1) 0 : テキストでぬりつぶす
LINEXE : 最終 X 座標 (X2) 1 : PSET 00~07H ノーマルボックスフル
LINEYE : 最終 Y 座標 (Y2) 2 : PRESET 08~7FH タイリングボックスフル
3 : XOR 80H TILBUF のパターンでボックスフル

PSMODE=0 のとき CHRCD, COLORF, KSENFG を参照します。また、GCOLOR が 80H のとき TILBUF を参照するので実行前に設定しておいて下さい。

ワークエリア SCRNM 3 ≥ 6 のとき何も実行しません。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
BOXSUB	5 6 0 4	LINEXS LINEYS LINEXE LINEYE PSMODE GCOLOR LINPAT		AF, BC, DE, HL, BC', DE', HL', IX, IY	四角形を描きます。(注1)
LINESB	5 6 9 F	LINEXS LINEYS LINEXE LINEYE PSMODE GCOLOR LINPAT		AF, BC, DE, HL, BC', DE', HL', IX, IY	直線を引きます。(注2)
ELHPUT	5 7 8 D	BC E L H	BC E L H	AF	PUT のルーチンです。(注3)
ELHGET	5 7 A A	BC : グラフィックア ドレス	BC E L H	AF, E, HL	GET のルーチンです。
PSETSB	5 7 F 1	PSETX : X座標 PSETY : Y座標 GCOLOR : パレットコ ード		AF, BC, DE, HL	PSET のルーチンです。
RESETS	5 8 0 C	PSETX : X座標 PSETY : Y座標 GCOLOR : パレットコ ード		AF, BC, DE, HL	PESET のルーチンです。
POINTS	5 8 B D	HL : X座標 DE : Y座標 SCRNM2 : スクリー ンモード (0…カラー 1, 2, 3…モノクロ)	A : パレットコード (0 ~ 7) CY = 1 ならウィンド ウ外	AF, BC, DE, HL	A = POINT(HL, DE)を行います。
GRAADR	5 9 0 7	HL : X座標 DE : Y座標 SCRNM2 : スクリー ンモード (0…カラー 1, 2, 3…モノクロ)	HL : グラフィックア ドレス CY = 1 ならウィンド ウ外	AF, BC, DE, HL	与えられたグラフィック座標が ウィンドウ内かチェックしてグ ラフィックアドレスを計算しま す。
GRAAD2	5 9 0 F	HL : X座標 DE : Y座標 SCRNM2 : スクリー ンモード (0…カラー 1, 2, 3…モノクロ)	HL : グラフィックア ドレス	AF, BC, DE, HL	グラフィックアドレスを計算し ます。ウィンドウのチェックは しません。
UPADR	5 9 A 8	BC : グラフィックア ドレス WK1FD0 WIDTH0 SCRNM3	BC : 1 ライン上げた あとのグラフィ ックアドレス	AF	グラフィックアドレスを1ライ ン分上げます。
DWADR	5 9 F C	BC : グラフィックア ドレス WK1FD0 WIDTH0 SCRNM3	BC : 1 ライン下げた あとのグラフィ ックアドレス	AF	グラフィックアドレスを1ライ ン分下げます。
CLSGRA	5 A 4 D	CLSMOD 0 : 青, 赤, 緑全て 1 : 青 2 : 赤 3 : 緑		AF, BC, DE, HL	グラフィック画面のクリアを行 います。
WINDOI	5 A D 8			AF, BC, DE, HL	ウィンドウを最大にします。

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
WINDST	5AEA	HL: X座標の最小値 DE: Y座標の最小値 HL: X座標の最大値 DE: Y座標の最大値		AF, BC, DE, HL, AF', BC', DE', HL'	パラメータを与えてウィンドウを設定します。
TILCOL	5B99	GCOLOR TILBUF GCOLOR \geq 80H のときは何も実行しません	TILBUF	AF, BC, DE, HL	タイルバッファにカラーパターンを設定します。
HPAINT	5EA1	PAINTX: X座標 PAINTY: Y座標 GCOLOR BKCLLN BKCOLOR TMPEND		AF, BC, DE, HL, BC', DE', HL'	任意の部分を指定したカラーでペイントします。(注4)
TILSET	61A5	A: バッファ No. (0~7) TILBUF: タイルパターン	TILLBF: タイルパターンデータ	AF, HL	タイルバッファにタイルパターンを設定します。
PATSUB	623D	GCURX: X座標 GCURY: Y座標 PATUDD DE A		AF, BC, BC', DE, DE', HL	PATTERN 処理ルーチンです。(注5)
POLYSB	630B	SINSX SINSY SINRX SINRY GCOLOR SIND SINSTA SINEND			多角形, または円・弧を描きます。(注6)
TEMPSB	656E	DE: テンポ (30~7500)		AF, BC, DE, HL	テンポを設定します(CTC3 カウンター設定 DE/7500)。
MUSICS	65AC	DE HL A		AF, BC, DE, HL	音楽演奏をします。(注7)

注1: LINE(X1, Y1)-(X2, Y2), モード, パレットコード, B, LINPAT の働きをします。ウィンドウからはみでた部分は、無視します。

LINPAT 2 バイトのラインパターン

SCRNM 3 \geq 6 のとき何も実行しません。

PSMODE=0 のとき CHRCOD, COLORF, KSENF, LINPAT を参照しますので実行前に設定しておいて下さい。

BOXFUL 参照

注2: LINE(X1, Y1)-(X2, Y2), モード, パレットコード, B, LINPAT の働きをします。ウィンドウからはみでた部分は、無視します。

SCRNM 3 \geq 6 のとき何も実行しません。

BOXFUL, BOXSUB 参照

注3: BC: グラフィックアドレス グラフィックモード 0 0000H~3FFFH E 青のデータ

1 4000H~7FFFH L 赤のデータ

2 8000H~BFFFH H 緑のデータ

3 C000H~FFFFH

注4: BKCOLOR は境界色の色番号を示します。境界色は複数指定できますので、その色数を BKCLLN で指定して下さい。TMPEND はペイント用ワークエリア先頭アドレスで、TMPEND からスタックポインタが示すアドレスまでがペイント用ワークエリアとして使用されます。

注5: PATUDD はタイルパターンの長さで 00H~7FH のとき下方向, 80H~FFH で上方向を意味します。DE でパターンデータの格納先頭アドレス, A でパターンデータの長さを設定してください。

注6: SINSX: 中心の X 座標 SINRX: 水平方向の半径 GCOLOR: パレットコード

SINSY: 中心の Y 座標 SINRY: 垂直方向の半径 SIND: ステップ角 360/n 度

SINSTA: 初期角 SINEND: 終了角

注7: DE は音楽データの格納先頭アドレス, HL のインターラプトバッファの先頭アドレスです。

A にはバックグラウンドプレイをするかしないかを示すミュージックモードの値が入ります。

A=1 のときバックグラウンドプレイをします。

なお音楽データは、アスキーコード文字列+エンドコード 00H で格納して下さい。

インターラプトバッファには、インターラプトジョブ用に変換されたデータが格納されます。

A=0 のとき DE=HL です。

インターラプトジョブ用データについては、MUBFST(65F2H)を参照してください。

付録B turboシリーズ

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
MUBFST	6 5 F 2	DE HL	DE HL A	AF, C	音楽データをインターラプトジョブ用データに変換します。(注1)
SIOCTC	6 D 3 F			AF, BC, DE, HL	CTC と SIO のイニシャライズを行います。(注2)
RSINIT	6 D A 5	H : CTC 1 データ L : SIO R4 データ D : SIO R5 データ E : SIO R3 データ		AF, BC, DE, HL	SIO のチャンネル A のモードセットを行います。(注2)
RXINP	6 E 5 9		A : 入力したデータ	AF	RS-232C インターフェイスよりデータの入力を行います。(注3)
RXSNS	6 E 8 3		ZF=1 データなし ZF=0 入力可能	AF, HL	RS-232C インターフェイスからのデータの入力ができるかどうかを調べます。
TXOUT	6 E 8 A	A : 出力するデータ	A : 出力したデータ	フラグ	RS-232C インターフェイスへデータの出力を行います。(注4)
TXSNS	6 E A 7		ZF=1 出力不可能 ZF=0 出力可能	AF, BC	RS-232C インターフェイスへデータが出力できるかどうか調べます。
MOUSE0	6 E A F			AF, BC	マウス割り込みモードの解除を行います。(注5)
MOUSE1	6 E C 0	HL : マウスカーソル 初期X座標 DE : マウスカーソル 初期Y座標		AF, BC, DE, HL	マウス割り込みモードの設定を行います。(注6)
SAVE1	7 0 2 0	HL : FCB 先頭アドレス DE : FCB サイズ	HL : FCB 先頭アドレス DE : FCB サイズ A : エラーコード	AF, BC	データレコーダへの FCB の出力を行います。(注7)
SAVE2	7 0 2 4	HL : データ先頭アドレス DE : データサイズ	HL : データ先頭アドレス DE : データサイズ A : エラーコード	AF, BC	データレコーダへのデータの出力を行います。(注8)
LOAD1	7 0 4 7	HL : FCB 先頭アドレス DE : FCB サイズ	HL : FCB 先頭アドレス DE : FCB サイズ A : エラーコード	AF, BC	データレコーダから FCB の入力を行います。(注8)
LOAD2	7 0 4 B	HL : データ先頭アドレス DE : データサイズ	HL : データ先頭アドレス DE : データサイズ A : エラーコード	AF, BC	データレコーダからデータの入力を行います。(注8)
VERFY2	7 0 5 C	HL : データ先頭アドレス DE : データサイズ	HL : データ先頭アドレス DE : データサイズ A : エラーコード	AF, BC	データレコーダへ出力したデータとメモリの内容を比較します。(注8)
CMTCOM	7 2 C 3	A : コントロールコード	A : コントロールコード		データレコーダのコントロールコードを出力します。(注9)
CMTSNS	7 2 C D		A	AF	データレコーダの状態を調べます。(注10)
FDCRED	7 3 9 D	HL : データ格納先頭アドレス DE : レコード番号 (読み込み開始) A FDCNO UNITNO		AF, BC, DE, HL, AF	デバイスよりデータの入力を行います。(注11)

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
FDCWRT	7 3 A A	HL: データ格納先頭アドレス DE: レコード番号 (読み込み開始) A FDCNO UNITNO		AF, BC, DE, HL, AF'	デバイスヘデータの出力を行います。(注12)
FDCVIFY	7 3 B 7	HL DE A FDCNO UNITNO		AF, BC, DE, HL, AF'	デバイスのベリファイを行います。(注13)
DSKRED	7 6 C A	HL D E A' UNITNO SECMIN SECMAX	HL: 次のデータ格納先頭アドレス D: 次のトラック番号 E: 次のセクタ番号	AF, BC, DE, HL, AF'	DMAを使用しないで3インチ or 5 インチディスクからデータの入力を行います。(注14)
DSKWRT	7 6 D 5	HL D E A' UNITNO SECMIN SECMAX	HL: 次のデータ格納先頭アドレス D: 次のトラック番号 E: 次のセクタ番号	AF, BC, DE, HL, AF'	DMAを使用しないで3インチ or 5 インチディスクヘデータの出力を行います。
DSKVIFY	7 6 E 0	HL D E A' UNITNO SECMIN SECMAX	HL: 次のデータ格納先頭アドレス D: 次のトラック番号 E: 次のセクタ番号	AF, BC, DE, HL, AF'	DMAを使用しないで3インチ or 5 インチディスクのベリファイを行います。
MOTOF8	7 7 9 2	MTOFIO UNITNO: ドライブNo.		AF, BC	3 インチ or 5 インチ or 8 インチディスクドライブのモータをOFFにします。(注15)

注1: 入力 of DE は、音楽データ格納先頭アドレスで、HL は、インターラプトジョブ用データ格納先頭アドレスです。出力 of DE は音楽データエンドアドレス+1 で、HL はインターラプトジョブ用データエンドアドレス+1 です。また出力 of A はエンドコードです。

注2: HL, DE にセットする値によってボーレート、パリティ、データビット長、ストップビット長が設定されます。

注3: このルーチンは、データの入力があるまでループしますので RXSNS(6E83H) で入力チェックを行ってからコールして下さい。

注4: このルーチンも RXINP(6E59H) と同様に出力があるまでループしますので TXSNS(6EA7H) で出力できるかチェックしてからコールして下さい。

注5: CTC0 にノンインターラプトモードが設定されることになります。

注6: CTC0, CTC2, SIOB にインターラプトモードが設定され、マウスを操作すれば関連したワークエリアが変化します。

注7: A: エラーコード

A=0 OK A=1 ブレイクされた
A=3 テープなし A=4 ライトプロテクトされている A=5 テープエンド

注8: エラーコードは SAVE1 と同じです。

注9: A: コントロールコード

A=0 EJECT A=1 STOP A=2 PLAY A=3 早送り A=4 巻きもどし
A=5 APSS1 A=6 APSS-1 A=10 REC

注10: A に入っているデータの bit2=1 だとプロテクトされており、bit1=1 だとテープがセットされており、bit0=1 だとデータレコーダ作動中ということです。

注11: DE には読み込みを開始するレコード番号、A にはレコード数、FDCNO にはデバイス No. UNITNO にはドライブ No. が入ります。

FDCNO=5 MEM0:~MEM1: FDCNO=6 EMM1:~EMM9: FDCNO=7 0:~3:(3 インチ or 5 インチディスク)

FDCNO=8 F0H~F3H(8 インチディスク) FDCNO=9 HD0:~HD3:(ハードディスク)

注12: FDCRED(739DH) を参照して下さい。

注13: HL, DE, A, FDCNO, UNITNO は FDCRED(739DH) を参照して下さい。ただし FDCNO は、5, 6, 7 の DMA を使用しないタイプだけしか指定できません。

注14: HL はデータ格納先頭アドレス、D=トラック No.×2+サイド No., E=セクタ No.(1~16), A'=セクタ数, UNITNO=ドライブ No., SECMIN:01H, SECMAX:10H が初期値です。

注15: MTOFIO=0FCH 3 インチ or 5 インチ, MTOFIO=0FCH 8 インチ

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
MOTOFF	7 7 9 7	UNITNO:ドライブNo.		AF BC	3 インチ or 5 インチディスクのモータを OFF にします。
HDINIT	7 8 D 9	A:ドライブNo. (0~0FH)		AF, HL, BC, IY, DE	ハードディスクのイニシャライズを行います。
HDOFFS	7 8 E 2	A:ドライブNo. (0~0FH)			BASICのHDOFFと同じ働きをします。(注1)
J PBCNE	7 D 6 C	BC:ROMアドレス	CY	フラグ	BCレジスタの示すROM内ルーチンにジャンプします。ただしエラー処理は行いません。(注2)
BIOSER	F 8 3 C	A:エラーコード SP, SP+1: エラー発生アドレス	A:エラーコード SP, SP+1: エラー発生アドレス	B	BIOSでエラーが発生したときの処理を行います。(注3)
KEYRAM	F 8 4 3				キー割り込みが行われた場合の処理を行います。(注4)
INTSUB	F 8 4 7				割り込み処理ルーチンへのジャンプを行います。(注5)
OPENF9	F 8 B 7		CY=1 エラー発生 CY=0 オープン可能	AF, BC, DE, HL	ユーザー辞書モードのオープン処理を行います。(注6)
OPENF8	F 8 B A		CY=1 エラー発生 CY=0 オープン可能	AF, BC, DE, HL	システム辞書モードのオープン処理を行います。
OPENF7	F 8 B D		CY=1 エラー発生 CY=0 オープン可能	AF, BC, DE, HL	音訓辞書モードのオープン処理を行います。
FINDF7	F 8 C 0	(注5)	(注5)	・システム/ ユーザー辞書 モード AF, BC, DE, HL ・音訓辞書モード AF, BC, DE	かな漢字変換処理を行います。(注7)
NEXTF7	F 8 C 3	(注1)	(注1)	・システム/ ユーザー辞書 モード AF, BC, DE, HL ・音訓辞書モード AF, BC, DE	次候補漢字のバッファを設定します。(注8)
BACKF7	F 8 C 6	(注1)	(注1)	・システム/ ユーザー辞書 モード AF, BC, DE, HL ・音訓辞書モード AF, BC, DE	前候補漢字のバッファを設定します。(注8)
X1CLF7	F 8 C 9	C HL		AF, BC, DE, HL	漢字の選択と学習機能処理を行います。(注9)
NEXTJS	F 8 C C		DE:シフトJISコード	AF, DE	音訓辞書モード次候補、シフトJISコード入力の処理を行います。
BITDES	F 8 D E	LPTBUF:データ LPOS B:データ長 KANJI F:モード		AF, BC, DE, HL	ビットイメージLPRINT用バッファの出力を行います。
HCOPYS	F 8 E 1	A:HCOPYモード		AF, BC, DE, HL	HCOPYの処理を行います。(注10)

ルーチン名	アドレス	入 力	出 力	破壊されるレジスタ	機 能
CPSM23	F8E4	A: HCOPIY モード		AF, BC, DE, HL	HCOPIY の処理を行います (WIDTH 20 or WIDTH 10). (注10)
KEYSNN	F8E7	POINT1 POINT2 INBUF POINT3 INBUF ZF=0 なら KEYGET OK	POINT1 POINT2 POINT3 INBUF ZF=0 なら KEYGET OK	AF, BC, DE, HL	キーセンスの処理を行います (POINT1, POINT2, INBUF の内容を POINT3, INBUF へ送ります). (注11)
SCRRAM	F8EA	D: Y LENGTH E: X LENGTH HL: スタートアトリ ビュートアドレス		AF, BC, DE, HL	テキスト V-RAM スクロールを処理します。
HDDMAS	F929				ハードディスクのリード/ライトを行います。
DSKWKS	F96E				コマンドを送ったあと、ディスク 1 セクタリード/ライトを行います。
MEMEMM	F9A4				MEM: EMM: のリード/ライトを行います。
HLDECK	FA25	HL: データ先頭アドレス DE: データ先頭アドレス C: データ長	ZF=1 ならすべて一致	AF	HL, DE の示すアドレスからの値を C バイト比較します。
SETRES	FA39				PSET, PRESET, XOR の処理を行います。
SETMD	FA3D				PSET, PRESET, XOR, POINT1 の処理を行います。
RESMD	FA40				PSET, PRESET, XOR, POINT0 の処理を行います。

注1: ヘッドをレコード No. 9E4BH までシークします。

注2: CY=1 のとき A にエラーコードが入ります。

CY=0 のときエラーは発生しません。

注3: バンクをメインメモリ側に切換えエラー処理ルーチン (000DH) へジャンプします。

注4: HL をスタックに待避した後 F845H (LOW), F846H (HIGH) で指示されるアドレスを次の INTSUB に渡します。そのため F845H, F846H にはあらかじめ実行させたいルーチンのアドレスを設定しておく必要があります。また、ユーザー割り込みルーチンは 8000H 以降に置きます。

注5: このルーチンは、バンク状況を待避した上でバンクを ROM に切り換え、HL で示される割り込み処理ルーチンへ制御を渡します。処理が終了するとバンクを元に戻して割り込みを解除します。このため、このルーチンと呼ぶときには HL に割り込み処理ルーチンのアドレスをセットしておく必要があります。

注6: 辞書のオープンチェックとかな漢字変換用ルーチン (F8C0H~F8CEH) の設定を行います。OPEN F9 (F8B7H) ~NEXTJS (F8CCH) までのルーチンはすべてジャンプテーブルです。BASIC が起動されると設定されます。

注7: ● システム/ユーザー辞書モードでは、入力 DE=アスキーバッファ先頭アドレス、HL=漢字バッファ先頭アドレス (最大40バイト) 出力は CY=1 なら Not Found, ZF=1 なら 1 Word のみ Found, 漢字バッファ=漢字データです。
● 音訓辞書モードでは入力 DE=アスキーバッファ先頭アドレス、出力は CY=1 なら Not Found, ZF=1 なら DE のコードのみ Found, DE=最初の漢字シフト JIS コード

注8: ・システム/ユーザー辞書モードでは、入力 HL=漢字バッファアドレス (最大40バイト) 出力は漢字バッファ=漢字データです。

・音訓辞書モードでは、入力はなし、出力は DE=漢字シフト JIS コードです。

注9: C=セレクトキー-1, データ...0, 1, 2, 3, 4, 5, 6, 7, 8 (例えば 6 を押せば C=5), CR を押したとき 00FFH, HL=セレクト漢字データアドレスこれはシステム/ユーザー辞書モード、音訓辞書モードでは同じです。又、システム/ユーザー辞書モードではセレクト漢字データのエンドコードは、30H 以下です。

注10: A=FFH テキスト A=0 グラフィック全て A=1 G1
A=2 G2 A=3 G3 A=4 テキストとグラフィック全て

DI と EI 命令は使用してはいけません。

注11: かな漢字変換処理のとき使われます。

B-2 ワークエリア

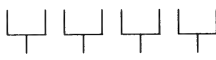
ラベル名	アドレス	内 容
FLMAX FLTEN FLONE FLLAST DLLMT SLMT	7 B 2 D 7 B A 5 7 B A D 7 B B 5 7 B B D 7 B C 5 7 B C A	浮動小数点データテーブル (1D+16・・100) [8 バイト] 浮動小数点データテーブル (10) 浮動小数点データテーブル (1) 浮動小数点データテーブル (0.1) 浮動小数点データテーブル (1D-16) 浮動小数点データテーブル (1E-8 [5 バイト], 1D-8)
WORK INTTAB	7 D 6 B 7 D 9 C F 8 0 0 F 8 3 B	関数用データテーブル ワークエリア (F800H~FA63H) のイニシャライズ用データ インタラプトジョブ・ジャンプ・テーブル * のテーブルは ROM では未使用です。ユーザーが自由に定義して使用することができます。 * F 8 0 0, 0 1 SIOB OUT BUFF EMPTY * F 8 0 2, 0 3 SIOB STATUS F 8 0 4, 0 5 SIOB IN BUFF OK (SIOBIN) F 8 0 6, 0 7 SIOB ERROR, SDLC (SIOBER) * F 8 0 8, 0 9 SIOA OUT BUFF EMPTY * F 8 0 A, 0 B SIOA STATUS F 8 0 C, 0 D SIOA IN BUFF OK (SIOAIN) F 8 0 E, 0 F SIOA ERROR, SDLC (SIOAER) * F 8 1 0, 1 1 DMA RDY INT * F 8 1 2, 1 3 DMA EQU INT * F 8 1 4, 1 5 DMA EOB INT * F 8 1 6, 1 7 DMA EOB EQU INT F 8 1 8, 1 9 CTC 0 (CTC0IN) 1msec COUNTER (MOUSE=64msec) F 8 1 A, 1 B キー・インタラプト・テーブル * F 8 1 C, 1 D CTC2 F 8 1 E, 1 F CTC3 (CTC3IN) (TEMPO) * F 8 2 0, 2 1 TIMER 0 * F 8 2 2, 2 3 TIMER 1 * F 8 2 4, 2 5 TIMER 2 * F 8 2 6, 2 7 TIMER 3 * F 8 2 8, 2 9 TIMER 4 * F 8 2 A, 2 B TIMER 5 * F 8 2 C, 2 D TIMER 6 * F 8 2 E, 2 F TIMER 7 * F 8 3 0, 3 1 ユーザーインタラプト用 * F 8 3 2, 3 3 ユーザーインタラプト用 * F 8 3 4, 3 5 ユーザーインタラプト用 * F 8 3 6, 3 7 ユーザーインタラプト用 * F 8 3 8, 3 9 ユーザーインタラプト用 * F 8 3 A, 3 B ユーザーインタラプト用
X1MODB	F 8 7 6	XFER モードバッファ bit 7 : 0・・ローマ字モード・オフ 1・・ローマ字モード bit 6 : 0・・全角 1・・半角 bit 5 : 0・・ひらがな 1・・カタカナ bit 4 : 0・・間接 1・・直接 bit 3 : 0・・コード入力 1・・変換 bit 3 が 1 のとき bit 2, 1 : 0 0・・一文字変換 0 1・・音訓辞書変換 1 0・・システム辞書変換 1 1・・ユーザー辞書変換 bit 3 が 0 のとき bit 0 : 0・・JIS 漢字またはシフト JIS コード 1・・区点コード
OPTKEY	F 8 7 7 F 8 7 8	ロールダウンキーのコード (0FH, 03H) ロールアップキーのコード (0EH, 04H)
HELPKY	F 8 7 9	ヘルプキーコード (11H)
COPYKY	F 8 7 A	コピーキーコード (10H)
GRAXMX	F 8 7 B	グラフィック X 座標最大値 (319, 639)
GRAYMX	F 8 7 D	グラフィック Y 座標最大値 (191, 199, 383, 399)

ラベル名	アドレス	内 容
WIDTH0	F 8 7 F	テキストX座標最大値+1 (40, 80)
CURVMX	F 8 8 0	テキストY座標最大値 (9, 11, 19, 24)
VRMGAI	F 8 8 1	外字 JIS 漢字コード(7621H~7660H)の上位バイトデータ(76H)
PRTGAI	F 8 8 2	プリンタ外字コード(7621H~7660H)の上位バイトデータ(76H)
LPCRCD	F 8 8 3	プリンタへのCR+LF出力モード (82H) ? 1 H . . . 0DH ? 2 H . . . 0AH ? 3 H . . . 0DH+0AH bit 7=1 のとき, HCOPY のCR+LF は OFF
LPPGCD	F 8 8 4	プリンタフォームフィードコード (0CH)
LPTGIC	F 8 8 5	HCOPY モード(640ドットモード)設定コード
		F 8 8 5 データ長: 02H F 8 8 6 データ長: 1BH F 8 8 7 データ長: 52H F 8 8 8 データ長: 00H F 8 8 9 データ長: 00H
LPTLSC	F 8 8 A	ラインフィードピッチコード(1キャラクタ) F 8 8 A データ長: 04H F 8 8 B データ長: 1BH F 8 8 C データ長: 25H F 8 8 D データ長: 39H F 8 8 E データ長: 0FH
LPTBTC	F 8 8 F	ビットイメージ・アウトコード+モード (03H, 1BH, 25H, 32H) + (00H) F 8 8 F データ長 bit 7 . . . BIT DATA TWICE モード = 1 のとき" A A B B C C D D" = 0 のとき" A B C D" F 8 9 0 } データ(最大3バイト) F 8 9 3 * F 8 9 0 + F 8 8 F LENGTH MODE (ビットイメージデータ長=上位*256+下位) 00H . . . CHR\$(上位)+CHR\$(下位) 01H . . . CHR\$(下位)+CHR\$(上位) 02H . . . USING"####";上位*256+下位
LPTLNC	F 8 9 4	ラインフィードピッチコード (02H, 1BH, 36H, 00H, 00H) F 8 9 4 データ長 F 8 9 5 } データ(最大4バイト) F 8 9 8
LPTGOC	F 8 9 9	ノーマルモード(960ドットモード)設定コード (02H, 1BH, 45H, 00H, 00H) F 8 9 9 データ長 F 8 9 A } データ(最大4バイト) F 8 9 D
LPTKIC	F 8 9 E	漢字 IN コードまたはビットイメージ LPRINT フラグ ●漢字プリンタの場合 F 8 9 E データ長 F 8 9 F } データ(最大3バイト) F 8 A 1 (例) CZ-80PK.....02H, 1BH, 4BH, 0 ●非漢字プリンタの場合 F 8 9 E 80H.....KMODE 1 のとき ビットイメージ KMODE 0 のとき アスキーコード LPRINT F0H.....KMODE 1, 0 とともビットイメージ F 8 9 F 1 ラインデータ最大値(80, 96, 120...最大140)

ラベル名	アドレス	内 容
DOTSPC	F 8 A 2	ドットスペースコード+オフセットまたはラインフィードピッチ設定コード (03H, 1BH, 25H, 39H, 00H) ●漢字プリンタの場合……漢字ドットスペースコード F 8 A 2 データ長 F 8 A 3 データ(最大3バイト) (例) CZ-80PK……(01H, 1BH)+(00H)+(0, 0) F 8 A 6 (注) データ長は実際に送るデータ長-1, 最終データはKLFTDTとKRGTDのオフセット ●非漢字プリンタの場合……ラインフィードピッチ設定コード F 8 A 2 データ長 bit 7=データイメージ 1…0 1 2 3 4 5 6 7 8 9 A B C D E F イメージ 0…0 8 1 9 2 A 3 B 4 C 5 D 6 E 7 F イメージ F 8 A 3 データ(最大4バイト) F 8 A 6
KLFTDT	F 8 A 7	漢字左ドットスペースまたは16ドットイメージ第1ラインフィードピッチコード(01H) ●漢字プリンタ 漢字左ドットスペースコード DOTSPC+KLFTDT…CHR\$ (&H1B, &H00+&H00) (例) CZ-80PK……00H ●非漢字プリンタ(01H) 第1ラインフィードピッチコード DOTSPC+KLFTDT…CHR\$ (&H1B, &H25, &H39, 1)
KRGTD	F 8 A 8	漢字右ドットスペースまたは16ドットイメージ第2ラインフィードピッチコード(17H) ●漢字プリンタ 漢字右ドットスペースコード DOTSPC+KRGTD…CHR\$ (&H1B, &H00+&H06) (例) CZ-80PK……06H ●非漢字プリンタ(17H) 第2ラインフィードピッチコード DOTSPC+KRGTD…CHR\$ (&H1B, &H25, &H39, &H17)
LPTKOC	F 8 A 9	漢字OUT コード (00H, 0, 0, 0) F 8 A 9 データ長 bit 7=LPTKIC, LPTKOC OUT MODE 1…ASCII; ASCII; KI; KANJI; KO; KI; KANJI; KO; ASCII; ASCII; … 0…ASCII; ASCII; KI; KANJI; KANJI; KO; ASCII; ASCII; …… F 8 A A データ(最大3バイト) F 8 A C
LPACHN	F 8 A D	漢字プリンタ半角コード上位バイトデータ (0FFH) 0FFH……半角コードではない 00H~0FEH……半角コード上位バイト ASCIIコード(20H~7FH, A0H~DFH)のとき, 漢字コード(???20H~???7FH, ??A0H~???DFH)
PRTDLY	F 8 A E	プリンタデバイスオフラインエラータイマーデータ (0DH)
LPTABL	F 8 A F	プリンタTAB幅データ (08H)
VRMPRS	F 8 B 0	データバッファプリンタデータ交換
RLARRA	F 8 B 2	プリンタヘッドのMSBまたはLSBセレクト ヘッドピン MSB……データ bit MSB (17H…RLA) データ bit LSB (1FH…RRA) ヘッドピン LSB……データ bit LSB (17H…RLA) データ bit MSB (1FH…RRA)

ラベル名		アドレス	内容
COLORF	F8D0	7トリビュートパツ77 (CSIZE, CGEN, CFLASH, CREV, COLOR) bit 7, 6 0 0CSIZE 0 0 1CSIZE 1 1 0CSIZE 2 1 1CSIZE 3 bit 5 0CGEN 0 1CGEN 1 bit 4 0CFLASH 0 1CFLASH 1 bit 3 0CREV 0 1CREV 1 bit 2, 1, 0 0 0 0COLOR 0 0 0 1COLOR 1 0 1 0COLOR 2 0 1 1COLOR 3 1 0 0COLOR 4 1 0 1COLOR 5 1 1 0COLOR 6 1 1 1COLOR 7	
	F8D1	スクリーンメモリキャラクタコード (20H)	
BPRIOF	F8D2	I/Oアドレス1000Hからパツ77 [パツ77ト.....青] G1=(4000H~7FFFH 1/O) G2=(8000H~BFFFH 1/O) G3=(C000H~FFFFH 1/O) bit 7 = 1...(G3=1, G2=1, G1=1 bit 6 = 1...(G3=1, G2=1, G1=0 bit 5 = 1...(G3=1, G2=0, G1=1 bit 4 = 1...(G3=1, G2=0, G1=0 bit 3 = 1...(G3=0, G2=1, G1=1 bit 2 = 1...(G3=0, G2=1, G1=0 bit 1 = 1...(G3=0, G2=0, G1=1 bit 0 = 1...(G3=0, G2=0, G1=0 ボイソト→青 ボイソト→青 ボイソト→青 ボイソト→青 ボイソト→青 ボイソト→青 ボイソト→青 ボイソト→青	
RPRIOF	F8D3	I/Oアドレス1100Hからパツ77 [パツ77ト.....赤] G1=(4000H~7FFFH 1/O) G2=(8000H~BFFFH 1/O) G3=(C000H~FFFFH 1/O) bit 7 = 1...(G3=1, G2=1, G1=1 bit 6 = 1...(G3=1, G2=1, G1=0 bit 5 = 1...(G3=1, G2=0, G1=1 bit 4 = 1...(G3=1, G2=0, G1=0 bit 3 = 1...(G3=0, G2=1, G1=1 bit 2 = 1...(G3=0, G2=1, G1=0 bit 1 = 1...(G3=0, G2=0, G1=1 bit 0 = 1...(G3=0, G2=0, G1=0 ボイソト→赤 ボイソト→赤 ボイソト→赤 ボイソト→赤 ボイソト→赤 ボイソト→赤 ボイソト→赤 ボイソト→赤	
GPRIOF	F8D4	I/Oアドレス1200Hからパツ77 [パツ77ト.....緑] G1=(4000H~7FFFH 1/O) G2=(8000H~BFFFH 1/O) G3=(C000H~FFFFH 1/O) bit 7 = 1...(G3=1, G2=1, G1=1 bit 6 = 1...(G3=1, G2=1, G1=0 bit 5 = 1...(G3=1, G2=0, G1=1 bit 4 = 1...(G3=1, G2=0, G1=0 bit 3 = 1...(G3=0, G2=1, G1=1 bit 2 = 1...(G3=0, G2=1, G1=0 bit 1 = 1...(G3=0, G2=0, G1=1 bit 0 = 1...(G3=0, G2=0, G1=0 ボイソト→緑 ボイソト→緑 ボイソト→緑 ボイソト→緑 ボイソト→緑 ボイソト→緑 ボイソト→緑 ボイソト→緑	

ラベル名	アドレス	内 容
TPRIOF	F8D5	I/Oアドレス1300H データバッファ (00H) [PRW データ] G1 = (4000H~7FFFH I/O) G2 = (8000H~BFFFH I/O) G3 = (C000H~FFFFH I/O) bit 7 = 1...(G3=1, G2=1, G1=1 ポイント)>テキスト bit 6 = 1...(G3=1, G2=1, G1=0 ポイント)>テキスト bit 5 = 1...(G3=1, G2=0, G1=1 ポイント)>テキスト bit 4 = 1...(G3=1, G2=0, G1=0 ポイント)>テキスト bit 3 = 1...(G3=0, G2=1, G1=1 ポイント)>テキスト bit 2 = 1...(G3=0, G2=1, G1=0 ポイント)>テキスト bit 1 = 1...(G3=0, G2=0, G1=1 ポイント)>テキスト bit 0 = 1...(G3=0, G2=0, G1=0 ポイント)>テキスト
WK1FD0	F8D6	I/Oアドレス1FD0H データバッファ (00H) bit 7 = 1...WIDTH, 20:WIDTH, 10 = 0...WIDTH, 25:WIDTH, 12 bit 6 = 1...CGREAD, 16ラインCG = 0...CGREAD, 8ラインCG bit 5 = 1...HIGH スピードCG アクセス = 0...X1 スピードCG アクセス bit 4 = 1...CPU アクセス G-RAM(バンク1) = 0...CPU アクセス G-RAM(バンク0) bit 3 = 1...CRT ディスプレイ G-RAM(バンク1) = 0...CRT ディスプレイ G-RAM(バンク0) bit 2 = 1...WIDTH, 12:WIDTH, 10 = 0...WIDTH, 25:WIDTH, 20 bit 1 = 1...G-RAM 1ライン=CRT 2ライン = 0...G-RAM 1ライン=CRT 1ライン bit 0 = 1...400ラインディスプレイモード = 0...200ラインディスプレイモード bit 7, 2, 1, 0 0 0 0 0 WIDTH, 25, 0, 1 0 0 0 1 WIDTH, 25, 1, 2 0 0 1 1 WIDTH, 25, 0, 2 0 1 0 0 WIDTH, 12, 0, 1 0 1 0 1 WIDTH, 12, 1, 2 0 1 1 1 WIDTH, 12, 0, 2 1 0 0 0 WIDTH, 20, 0, 1 1 0 1 1 WIDTH, 20, 0, 2 1 1 0 0 WIDTH, 10, 0, 1
SCRMOD	F8D7	オプションスクリーンモードデータ (04H) 0...バンク0=グラフィック バンク1=グラフィック 1...バンク0=グラフィック バンク1=変数 2...バンク0=ファイル(MEM0:) バンク1=変数 3...バンク0=グラフィック バンク1=FILE(MEM1:) 4...バンク0=ファイル(MEM0:) バンク1=FILE(MEM1:)
SECMIN	F8D8	セクターNo最小値 (01H)
SECMAX	F8D9	セクターNo最大値 (10H or 1AH) 3インチ or 5インチ or 8インチディスク NEXT SECTOR チェックワークエリア
PRCSO	F8DA	データタイプ (08H) 2.....整数型 3.....文字型 5.....単精度 8.....倍精度
REPTF1	F8DB	リピート ON/OFF データ (01H) 0.....リピートOFF 1.....リピートON
TMPEND	F8DC	BASIC テキスト, 変数, 文字変数, ワークエリアエンドアドレス (8000H) PAINT 用スタックワークエリアスタートアドレス
RAMCR1	F8ED	INPUTF, BINPUT CR 処理テーブル (0C3H, 078H, 017H).....CR+LF (0C9H, 078H, 017H).....カーソルポジションリターン エディットコマンドで使用
MTOFIO	F928	ディスクモーターOFF I/Oアドレスバッファ 0FCH.....3インチ or 5インチ 0ECH.....8インチ

ラベル名	アドレス	内 容
X1MDCL	FA 4 6	XFER モード・カラー (05H)
X1SLCL	FA 4 7	XFER セレクト・カラー (15H)
CLICKM	FA 4 8	クリック音データバッファ(8バイト)
INSPRT	FA 5 0	スクロールモードバッファ C3HBINPUT or INPUTF 11HACCPRT
POWERF	FA 5 1	サブルーチン POWERS(4AD9H)用データタイプバッファ (05H)
SEED	FA 5 2	サブルーチン RND(4E96H)用 SEED バッファ(2バイト)
MEMMAX	FA 5 4	LIMIT アドレスバッファ (0F800H)
H CXMIN	FA 5 6	HCOPY X座標最小値 (00H)
H CXMAX	FA 5 7	HCOPY X座標最大値 (27H)
H CYMIN	FA 5 8	HCOPY Y座標最小値 (00H)
H CYMAX	FA 5 9	HCOPY Y座標最大値 (31H or 2FH)
MOUSX1	FA 5 A	マウスウィンドウ X座標最小値 (0000H)
MOUSY1	FA 5 C	マウスウィンドウ Y座標最小値 (0000H)
MOUSX2	FA 5 E	マウスウィンドウ X座標最大値 (013FH)
MOUSY2	FA 6 0	マウスウィンドウ Y座標最大値 (00C7H)
MOUSXD	FA 6 2	マウスX方向移動ステップ-1 (0~63) (09H)
MOUSYD	FA 6 3	マウスY方向移動ステップ-1 (0~63) (09H)
TABBUF	FA 6 4	水平タブポジションデータバッファ(80バイト) (注) 1 のところがタブポジション カーソルX 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 : 0~15 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 : 16~31 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 : 32~47 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 : 48~63 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 : 64~79
FD5DRT	FAB 4	3 インチ or 5 インチディスクドライブタイプ(4バイト) FAB4 ドライブ0用データ FAB5 ドライブ1用データ FAB6 ドライブ2用データ FAB7 ドライブ3用データ データ 0...NON DMA 2D ドライブ(320K バイト) 1...DMA 2DD ドライブ(640K バイト) 2...DMA 2HD ドライブ(1M バイト ALL 256 バイト フォーマット) 3...DMA 2HD ドライブ(1M バイト 1トラック 1セクタ 1サイド 128) 4...DMA 2S ドライブ(140K バイト) 5...DMA 2D ドライブ(320K バイト)
FD8DRT	FAB 8	8 インチディスクドライブタイプ(1バイト) bit 7 6 5 4 3 2 1 0  ドライブ 3 2 1 0 データ 0 0.....2D-256 ドライブ(1M バイト ALL 256 バイトフォーマット) 0 1.....2D-256 ドライブ(1M バイト 1トラック 1セクタ 1サイド 128) 1 0.....1S-128 ドライブ(240K バイト ALL 128 1サイドのみ)
DMAIOF	FAB 9	DMA ディスクドライブ メイン RAM, I/O RAM 切り換えデータ (00H) 0 0 H...メイン RAM 読み込み, 書き込み 0 8 H...I/O RAM 読み込み, 書き込み
FUNADR	FABA	ファンクションキーデータバッファアドレステーブル (0000H) 0000H のとき未定義 それ以外は, ファンクションキーデータ格納先頭アドレス
FKYDSF	FABC	ファンクションキーディスプレイデータ (00H) 00H...表示しない 01H...表示する

ラベル名	アドレス	内 容
DIRIMG	FABD	ディレクトリバッファ(32バイト) FABD ファイルタイプ…… 1=Obj, 2=Bas, 4=Asc FABE } ファイルネーム(13バイト) FACA FACB } 拡張子(3バイト) FACD FACE パスワード FACF ファイル長(下位) FAD0 ファイル長(上位) FAD1 ファイルのスタートアドレス(下位) FAD2 ファイルのスタートアドレス(上位) FAD3 ファイル実行アドレス(下位) } Objのみ有効 FAD4 ファイル実行アドレス(上位) FAD5 } 年, 月, 日, 時, 分 FAD9 FADA ワーク FADB ファイル先頭クラスタ値(下位) FADC ファイル先頭クラスタ値(上位)
FDCNO	FADD	デバイスNoワークエリア 0……SCR: 1……CRT: 2……DEY: 3……LPT: 4……CAS: 5……MEM0:~MEM1: 6……EMM0:~EMM9: 7……0:~3:(3インチ or 5インチディスク) 8……F0:~F3:(8インチディスク) 9……HD0:~HD3:(ハードディスク) 0AH……COM:
UNITNO	FADE	ドライブNoワークエリア
CURX	FADF	カーソルXポジション
CURY	FAE0	カーソルYポジション
COPYXY	FAE1	コピーキーカーソルポジション FAE1 X位置 FAE2 Y位置
CURYST	FAE3	コンソールYスタートポジション
CURYED	FAE4	コンソールYエンドポジション
CURXST	FAE5	コンソールXスタートポジション
CURXED	FAE6	コンソールXエンドポジション
LPOSST	FAE7	プリンターXスタートポジション
LPOS LN	FAE8	プリンターX桁数
LPPAGE	FAE9	プリンターYポジション
LPPGST	FAEA	プリンターYスタートポジション
LPPGLN	FAEB	プリンターY行数
CLICKF	FAEC	クリック音ON/OFF データ 00H……ON 00H以外……OFF
KEYDAT	FAED	INKEY\$(0)データバッファ
KEYDA2	FAEE	INKEY\$(2)データバッファ
COU1MS	FAEF	CTC0 インターラプトカウンタバッファ
BRKBUF	FAF0	SHIFT+BREAK(03H), CTRL-S(13H)バッファ
ONKYBF	FAF1	ファンクションキーインターラプトデータ(ON KEY GOSUB) 0以外のとき, インターラプト
RSINTF	FAF2	RS-232C(COM:)インターラプトデータ 1のとき, インターラプト

ラベル名	アドレス	内 容
RSERRF	FAF 3	RS-232C(COM:)エラーデータ データ bit 5Rx オーバーラン bit 4パリティ bit 1バッファオーバー
INTFLG	FAF 4	ファンクションキーインターラプトデータ (ON KEY GOSUB) FAF4 KEY 1 バッファ bit 7, 6 FAF5 KEY 2 バッファ 1 1KEYn STOP FAF6 KEY 3 バッファ 1 0KEYn ON FAF7 KEY 4 バッファ 0 0KEYn OFF FAF8 KEY 5 バッファ FAF9 KEY 6 バッファ bit 4 FAFA KEY 7 バッファ 1KEY IN FAFB KEY 8 バッファ 0NORMAL FAFC KEY 9 バッファ FAFD KEY 10 バッファ
POINT1	FAFE	INBUF 書き込みポインター
POINT2	FAFF	INBUF 読み込みポインター
INBUF	FB 0 0	キーインターラプトデータバッファ(64バイト)
POINT3	FB 4 0	INPBUF 書き込みポインター
INPBUF	FB 4 1	INPBUF 読み込みポインター FB42 { キーセンスデータバッファ(40 バイト) FB69
INIADR	FB 6 A	CRTC ディスプレイオフセットアドレスワークエリア FB6A 上位 FB6B 下位 SCREEN 0 : SCREEN 200H, 00H SCREEN 1 : SCREEN 304H, 00H or 02H, 00H
INIADW	FB 6 C	CPU アクセスオフセットアドレスワークエリア FB6C 下位 FB6D 上位 SCREEN, 0 : SCREEN, 200H, 00H SCREEN, 1 : SCREEN, 300H, 04H or 00H, 02H
KANJIF	FB 6 E	KMODE データ 0KMODE 0 1KMODE 1
KBUFSW	FB 6 F	KBUF ON/OFF データ 0KBUF ON 0 以外...KBUF OFF
CSIZEF	FB 7 0	CSIZE データ 0CSIZE 0 1CSIZE 1 2CSIZE 2 3CSIZE 3
LPOSB	FB 7 1	LPOS(0)ワークエリア
LPOSA	FB 7 2	LPOS(1)ワークエリア
LPOSK	FB 7 3	LPOS(2) * 2 ワークエリア
FILOUT	FB 7 4	SCR:, LPT: 切り換えデータ 0SCR: 1LPT: (注) サブルーチン CRIPRP, ACCPRP, TABPRP で参照します
ESCFLG	FB 7 5	漢字第1バイトワークエリア (SCR: or CRT:)
ESCPRF	FB 7 6	漢字第1バイトワークエリア (LPT:)
CTRLAF	FB 7 7	INPUTF, BINPUT の CTRL-A 処理用データ 0ノーマルモード 1CTRL-A モード
KEYFLG	FB 7 8	インターラプトキーデータ 0 キーデータ不可能 FFH ... キーデータ可能
GRACOD	FB 7 9	CTRL+GRAPH+KEY モードグラフィックコードバッファ

ラベル名	アドレス	内 容
ROMFLG	FB 7 A	モニターコマンドデータ 0.....RAM アクセス 1.....ROM アクセス
CTRLMD	FB 7 B	CTRL+M or CTRL+J (BASIC or CP/M データ) 0.....BASIC 1.....CP/M
SIOBR5	FB 7 C	SIO B チャンネル R5 データバッファ
CHRAND	FB 7 D	RS-232C ビット長 (Rx データマスクデータ) 0FFH 8 ビット 07FH 7 ビット 03FH 6 ビット 01FH 5 ビット
MONSP	FB 7 E	モニタースタックポインタワークエリア (2 バイト)
CHADDR	FB 8 0	MUSIC チャンネル A データスタートアドレス FB 8 0 下位 FB 8 1 上位
MUADDR	FB 8 2	MUSIC チャンネル A データアドレス FB 8 2 下位 (0000H のとき END) FB 8 3 上位 FB 8 4 チャンネル A ボリューム (00H~10H) FB 8 5 チャンネル A 音譜の長さ 01H, 02H, 03H, 04H, 06H, 08H, 0CH, 10H, 18H, 20H.....データ C0 C1 C2 C3 C4 C5 C6 C7 C8 C9MUSIC コード FB 8 6 チャンネル A オクターブデータ D0H, E0H, F0H, 00H, 10H, 20H, 30H, 40H.....データ 01 02 03 04 05 06 07 08MUSIC コード
MUACOU	FB 8 7	チャンネル A カウンター (01H~20H)
CHBADR	FB 8 8	MUSIC チャンネル B データスタートアドレス FB 8 8 下位 FB 8 9 上位
MUBADR	FB 8 A	MUSIC チャンネル B データアドレス FB 8 A 下位 } 0000H のとき END FB 8 B 上位 } FB 8 C チャンネル B ボリューム FB 8 D チャンネル B 音譜の長さ FB 8 E チャンネル B オクターブデータ
MUBCOU	FB 8 F	チャンネル B カウンター (01H~20H)
CHCADR	FB 9 0	MUSIC チャンネル C データスタートアドレス FB 9 0 下位 FB 9 1 上位
MUCADR	FB 9 2	MUSIC チャンネル C データアドレス FB 9 2 下位 } 0000H のとき END FB 9 3 上位 } FB 9 4 チャンネル C ボリューム FB 9 5 チャンネル C 音譜の長さ FB 9 6 チャンネル C オクターブデータ
MUCCOU	FB 9 7	チャンネル C カウンター (01H~20H)
MOUSEX	FB 9 8	マウス X 座標ワークエリア FB 9 8 下位 FB 9 9 上位
MOUSEY	FB 9 A	マウス Y 座標ワークエリア FB 9 A 下位 FB 9 B 上位
MOUSEF	FB 9 C	未使用
MOUPNT	FB 9 D	マウスデータインプットデータ
MOUDAT	FB 9 E	マウスデータインプットバッファ (3 バイト)
MS1OFX	FBA 1	マウススイッチ 1 OFF X 座標ワークエリア FBA 1 下位 FBA 2 上位

ラベル名	アドレス	内 容
MS1OFY	FBA3	マウススイッチ1 OFF Y座標ワークエリア FBA3 下位 FBA4 上位
MS1ONX	FBA5	マウススイッチ1 ON X座標ワークエリア FBA5 下位 FBA6 上位
MS1ONY	FBA7	マウススイッチ1 ON Y座標ワークエリア FBA7 下位 FBA8 上位
MS2OFX	FBA9	マウススイッチ2 OFF X座標ワークエリア FBA9 下位 FBAA 上位
MS2OFY	FBAB	マウススイッチ2 OFF Y座標ワークエリア FBAB 下位 FBAC 上位
MS2ONX	FBAD	マウススイッチ2 ON X座標ワークエリア FBAD 下位 FBAE 上位
MS2ONY	FBAF	マウススイッチ2 ON Y座標ワークエリア FBAF 下位 FBB0 上位
RSSTCT	FBB1	RS-232C XON/XOFF データ 0 0 H.....RTS コントロール 0 1 H.....NON コントロール 以外.....CTRL-S, CTRL-Q コントロール
RSPNT1	FBB2	RS-232C RSBUFF 書き込みポインター
RSPNT2	FBB3	RS-232C RSBUFF 読み込みポインター
RSBUF	FBB4	RS-232C インターラプトデータバッファ(64バイト)
SCRN00	FBF4	スクリーンアクセスモードデータ 0.....SCREEN 0 1.....SCREEN 1
SCRN01	FBF5	スクリーンアウトモードデータ 0.....SCREEN, 0 1.....SCREEN, 1
SCRNM2	FBF6	スクリーングラフィックカラー/モノクロモードデータ 0.....カラー 1.....青のみ 2.....赤のみ 3.....緑のみ
SCRNM3	FBF7	WIDTH, L, M, Dモードワークエリア1 00H.....WIDTH, 25, 0, 1 01H.....WIDTH, 12, 0, 1 02H.....WIDTH, 25, 0, 2 03H.....WIDTH, 12, 0, 2 04H.....WIDTH, 25, 1, 2 05H.....WIDTH, 12, 1, 2 06H.....WIDTH, 20, 0, 1 07H.....WIDTH, 10, 0, 1 08H.....WIDTH, 20, 0, 2
SCRNM4	FBF8	WIDTH, L, M, Dモードワークエリア2 bit 7, 6, 5, 4, CRTディスプレイセレクト(WIDTH, ..., D) 0 0 0 0 D=0 0 0 0 1 D=1 0 0 1 0 D=2 bit 3, 2, 1, 0, スクリーンモード(WIDTH, L, M) 0 0 0 0 25, 0 0 0 0 1 12, 0 0 0 1 0 20, 0 0 0 1 1 10, 0 0 1 0 0 25, 1 0 1 0 1 12, 1

ラベル名	アドレス	内 容
KSENF9	FBF9	アンダーラインモードデータ 00H.....KSEN 0 20H.....KSEN 1
INTMUF	FBFA	MUSICS モードデータ 0ノーマル MUSIC 1インタラプト MUSIC
VFLAG	FBFB	ウィンドウタイプデータ 0WINDOW(X1, Y1)-(X2, Y2) 1WINDOW(X1, Y1)-(X2, Y2), (X3, Y3)-(X4, Y4)
GCURXS	FBFC	ウィンドウX始点座標
GCURYS	FBFE	ウィンドウY始点座標
GCURXE	FC00	ウィンドウX終点座標
GCURYE	FC02	ウィンドウY終点座標
WIBYXS	FC04	ウィンドウXスタートバイトポイント(0~79)
WIBIXS	FC05	ウィンドウXスタートビットポイント(0~7)
WIBYXE	FC06	ウィンドウXエンドバイトポイント(0~79)
WIBIXE	FC07	ウィンドウXエンドビットポイント(0~7)
CLSECD	FC08	グラフィックCLS Xエンドビットポイント
CLSFGD	FC09	グラフィックCLS Xスタートビットポイント
CLSXS	FC0A	グラフィックCLS Xスタートバイトポイント
CLSXLN	FC0B	グラフィックCLS Xバイト LENGTH
CLSYLN	FC0C	グラフィックCLS Yライン LENGTH(2バイト)
SCRNXS	FC0E	ウィンドウチェック用ワークエリア(2バイト) SCRNXS=0-GCURXS
SCRNXE	FC10	ウィンドウチェック用ワークエリア(2バイト) SCRNXE=GCURXS-GCURXE-1
SCRNYS	FC12	ウィンドウチェック用ワークエリア(2バイト) SCRNYS=0-GCURYS
SCRNYE	FC14	ウィンドウチェック用ワークエリア(2バイト) SCRNYE=GCURYS-GCURYE-1
GCOLOR	FC16	グラフィックカラー 00H~07H.....カラー 08H~7FH.....タイルカラー 80H.....タイルパターン
GETXS LINEXS PSETX	FC17	ラインX始点座標(下位, 上位) PSET, PRESET X座標(下位, 上位)
GETYS LINEYS PSETY	FC19	ラインY始点座標(下位, 上位) PSET, PRESET Y座標(下位, 上位)
GETXE LINEXE	FC1B	ラインX終点座標(下位, 上位)
GETYE LINEYE	FC1D	ラインY終点座標(下位, 上位)
GCURX	FC1F	ポジションXワークエリア FC1F 下位 FC20 上位
GCURY	FC21	ポジションYワークエリア FC21 下位 FC22 上位
SCRNT0	FC23	スクリーン0 テキストV-RAM コネクタデータ(26バイト)
SCRNT1	FC3D	スクリーン1 テキストV-RAM コネクタデータ(26バイト) 0先頭行 1継続行

ラベル名	アドレス	内 容
SCRNTC	FC 5 7	SCRNT0 or SCRNT1 ワークセレクトテーブル(2バイト)
DSKTRK	FC 5 9	3 インチ or 5 インチディスクトラックワークエリア FC 5 9 ドライブ0トラック FC 5 A ドライブ1トラック FC 5 B ドライブ2トラック FC 5 C ドライブ3トラック
DSK8TK	FC 5 D	8 インチディスクトラックワークエリア FC 5 D ドライブ0トラック FC 5 E ドライブ1トラック FC 5 F ドライブ2トラック FC 6 0 ドライブ3トラック
DKIOSW	FC 6 1	ディスク 2DD or 2HD セレクトワークエリア 0FFH.....2DD 0FEH.....2HD
COMLIN	FC 6 2	モニタープロンプトマークワークエリア *(2AH).....RAM アクセスモード) (29H).....ROM アクセスモード
DSKERR	FC 6 3	ディスクエラーステータスワークエリア bit 7 = 1準備されていない bit 6 = 1ライトプロテクト bit 4 = 1フォーマットされていない(アンフォーマットディスク) bit 3 = 1CRC エラー bit 2 = 1データ無効
SCRLAD	FC 6 4	スクロールワークエリア(2バイト)
SUMDT	FC 6 6	ロード, ベリファイチェックサムワークエリア(2バイト)
TIMBUF	FC 6 8	TIME = 0秒数(5バイト PRCSON = 5)
LPTBUF	FC 6 D	ビットイメージ LPRINT 用 1 ラインバッファ(140バイト)
H1RAFL	FC F 9	ひらがな/カタカナデータ 0ひらがな 1カタカナ
KANBUF	FC FA	カナ漢字変換(XFER モード) 1 ラインバッファ(60バイト)
ONEBUF	FD 3 6	一文字変換漢字コードバッファ(2バイト)
ONESTA	FD 3 8	一文字変換漢字 1st コードバッファ(2バイト)
ONEEND	FD 3 A	一文字変換漢字 LAST コードバッファ(2バイト)
HENBUF	FD 3 C	カナ漢字変換(XFER モード)変換バッファ(41バイト)
HENASC	FD 6 5	カナ漢字変換 変換アスキーコードバッファ(11バイト)
X1HELP	FD 7 0	カナ漢字変換 HELP モードデータ 0ノーマルモード 1ヘルプモード
X1FUNC	FD 7 1	カナ漢字変換 ファンクションキーモードデータ 0モードチェンジャー 1ファンクションキー
X1MODE	FD 7 2	XFER モードバッファ bit 7 : 0ローマ字モード OFF : 1ローマ字モード ON bit 6 : 0全角 : 1半角 bit 5 : 0ひらがな : 1カタカナ bit 4 : 0間接 : 1直接 bit 3 : 0コード入力 : 1変換 bit 3 = 1 のとき bit 2, 1 0 0 一文字 0 1 音訓 1 0 システム辞書 1 1 ユーザー辞書 bit 3 = 0 のとき bit 0 : 0JIS 漢字コードまたはシフト JIS コード : 1区点コード

ラベル名	アドレス	内 容
X1POS	FD73	XFER モードXポジション-8
X1ESCF	FD74	XFER モード漢字1st バイトワークエリア
RMAASC	FD75	XFER モードローマ字コードバッファ(4バイト)
RMAKAN	FD79	XFER モードローマ字コード→カタカナコード 変換バッファ(4バイト)
COPYMD	FD7D	HCOPY モードワークエリア(2バイト)
HCOPYB	FD7F	HCOPY or ビットイメージLPRINT データ切り換えバッファ(24バイト)
DAYMES	FD9F	DATE\$, TIME\$, DAY\$メッセージワークエリア(8バイト+DATEBF 1バイト)
DATEBF	FDA7	DATE\$, DAY\$ 読み込み/書き込みワークエリア(3バイト)
DAYBF	FDA8	日付ワークエリア
TIMEBF	FDAA	TIME\$読み込み/書き込みワークエリア(3バイト)
NESTAK	FDAD	JPBCNE用スタックワークエリア(2バイト)
HDBORD	FDAF	HD ボードセレクトワークエリア(1バイト)
	FDB0	HD ドライブセレクトワークエリア(1バイト) bit 0~7.....ドライブ0~7 n=0.....未使用 n=1.....使 用
CMDTBL	FDB1	HD コマンドデータテーブル(6バイト)
HDDRV	FDB2	HD ドライブ0 or 1セレクト(00H or 20H)
HDREC	FDB3	HD レコードNo. FDB3 上位 FDB4 下位
HDLEN	FDB5	HD 読み込み/書き込みレコード長 FDB6 データ00H
HDSPCB BCOUNT	FDB7	アサインディスクパラメータワークエリア(10バイト) DBLMUL 用ワークエリア(1バイト)
CYFLG	FDB8	DBLMUL 用ワークエリア(1バイト)
ZFAC	FDB9	テンポラリーFAC(8バイト)
ZFAC1	FDC1	テンポラリーFAC(8バイト)
ZFAC2	FDC9	テンポラリーFAC(8バイト)
DGITCO	FDD1	CONV ワークエリア(1バイト)
DGITFG	FDD2	CONV ワークエリア(1バイト)
EXPFLG	FDD3	CONV ワークエリア(1バイト)
PRODFL	FDD4	CONV ワークエリア(1バイト)
DGBFM8 DGBFM1 DGBF00 DGBF08 DGBF10 DGBF11 DGBF12 DGBF16 DGBF17	FDD5 FDDC FDDD FDE5 FDE7 FDE8 FDE9 FDED FDEE	CONV アスキーデータワークエリア(以下DGBF17までトータル58バイト)
CLIPX1 CLIPY1 CLIPX2 CLIPY2 WINDX1 WINDY1 WINDX2 WINDY2	FDD5 FDD7 FDD9 Fddb FDDD FDDF FDE1 FDE3	クリッピングワークエリア(WINDY2まで各2バイト)
\$0FE00	FE00	IPL ディスクFCB ワークエリア(256バイト)
FATBUF	FE00	BASIC ディスクFAT ワークエリア(256バイト)

ラベル名	アドレス	内 容
SNFAC 0	FE 0 F	FAC アドレスワークエリア 0 FE 0 F 下位 FE 1 0 上位
SNFAC 1	FE 1 1	FAC アドレスワークエリア 1 FE 1 1 下位 FE 1 2 上位
SNFAC 2	FE 1 3	FAC アドレスワークエリア 2 FE 1 3 下位 FE 1 4 上位
SNFAC 3	FE 1 5	FAC アドレスワークエリア 3 FE 1 5 下位 FE 1 6 上位
SNFAC 4	FE 1 7	FAC アドレスワークエリア 4 FE 1 7 下位 FE 1 8 上位
SNFAC 5	FE 1 9	FAC アドレスワークエリア 5 FE 1 9 下位 FE 1 A 上位
EXPSIN	FE 1 B	EXP[HL]ワークエリア(1バイト)
EXPOFF	FF 1 C	EXP[HL]ワークエリア(1バイト)
EXPHBT	FE 1 D	EXP[HL]ワークエリア(1バイト)
LOGEXP	FE 1 E	LOG[HL]ワークエリア(1バイト)
SINSGN	FE 1 F	SIN[HL], COS[HL]ワークエリア(1バイト)
TILBUF	FE 2 0	タイルパターンバッファ(24バイト) (注)青 8バイト 赤 8バイト 緑 8バイト
BAKBUF	FE 3 8	HPAINT 用バックカラーパターンバッファ (16バイト+BKCOLR 8バイト)
BKCOLR	FE 4 8	HPAINT 境界色(最大8バイト), DB...0, 1, 2, 3, 4, 5, 6, 7
BKCLLN	FE 5 0	HPAINT 境界色データ数(0~8)
CHRCOD	FE 5 1	キャラクタライン, キャラクタ塗りつぶし, キャラクタアスキーコード (LINESB, BOXSUB, BOXFUL)
CLSMOD	FE 5 2	グラフィック CLS パラメータ 00H, 04HALL 01H青のみ 02H赤のみ 03H緑のみ
PUTMOD PSMODE	FE 5 3	LINESB, BOXSUB, BOXFUL, SYMB SB モードバッファ 00Hキャラクター 01HPSET 02HPRESET 03HXOR
LINPAT	FE 5 4	LINESB, BOXSUB ラインパターンデータ(2バイト)
PATUDD	FE 5 6	PATSUB 積み重ね段数データ(1バイト) 01H~7FHデータ UP 80H~FFHデータ DOWN
POLCIR	FE 5 8	POLY / CIRCLE セレクトデータ ※ BASIC で使用, ROM では使用しない。 00HPOLY 01HCIRCLE
PAINTX	FE 5 9	ペイント X 座標バッファ FE 5 9 下位 FE 5 A 上位
SINSX	FE 5 9	CIRCLE / POLY 中心 X 座標バッファ SYMBOL X 座標バッファ FE 5 9 下位 FE 5 A 上位

ラベル名	アドレス	内 容
PAINTY	FE 5 B	ペイントY座標バッファ FE 5 B 下位 FE 5 C 上位
SINSY	FE 5 B	CIRCLE / POLY 中心Y座標バッファ SYMBOL Y座標バッファ FE 5 B 下位 FE 5 C 上位
SINRX	FE 5 D	CIRCLE / POLY X方向半径 FE 5 D 下位 FE 5 E 上位
SINRY	FE 5 F	CIRCLE / POLY Y方向半径 FE 5 F 下位 FE 6 0 上位
SIND	FE 6 1	CIRCLE / POLY ステップ FE 6 1 下位 FE 6 2 上位
		SYMBOL 角度方向(0, 1, 2, 3)
GETADR SINSTA	FE 6 3	SYMBOL データスタートアドレス CIRCLE / POLY 初期角 FE 6 3 下位 FE 6 4 上位
ARYEDA SINEND	FE65	CIRCLE / POLY 終了角 FE 6 5 下位 FE 6 6 上位
XMULHI XMULLO YMULHI YMULLO SINXAD SINYAD ENTPY LX LP LA RX RA OLX OLA ORX ORA OOLX OOLA OORX OORA CHKX STFLAG LKFLAG STKTOP STKBTM NXTPSH NXTPOP TILLBF XINC0 XINC1 YINC2 YINC3 XMOD	FE 6 7 FE 6 8 FE 6 A FE 6 B FE 6 D FE 6 F FE 7 1 FE 7 3 FE 7 4 FE 7 5 FE 7 7 FE 7 9 FE 7 B FE 7 D FE 7 F FE 8 1 FE 8 3 FE 8 5 FE 8 7 FE 8 9 FE 8 B FE 8 D FE 8 E FE 8 F FE 9 1 FE 9 3 FE 9 5 FE 9 7 FE 9 A FE 9 C FE 9 E FE A 0 FE A 2	ライン用ワークエリア YMOD までグラフィックルーチン用ワークエリア ペイントワークエリア SYMBOL X方向倍率(下位, 上位) SYMBOL Y方向倍率(下位, 上位)
YMOD	FE A 4	
ZZZZZZ	FE A 6	グラフィックワークエリアエンド
DIRBUF	FF 0 0	BASIC FILES バッファ(256バイト)
KEYBUF	FF 0 0	ライン入力キーバッファ(256バイト)

ラベル名	アドレス	内 容
\$0FF00	FF00	IPL用FCBバッファ
\$0FF01	FF01	・ファイルタイプ
\$0FF12	FF12	・ファイルネーム
\$0FF14	FF14	・ファイルの長さ
\$0FF16	FF16	・ロードアドレス
\$0FF18	FF18	・実行アドレス
\$0FF19	FF19	・年
\$0FF1A	FF1A	・月, 曜日
\$0FF1B	FF1B	・日
\$0FF1D	FF1D	・時, 分
\$0FF1E	FF1E	・秒
		・スタートレコードNo.(下位, 上位)
\$0FF20	FF20	IPLタイマーエリア
\$0FF26	FF26	IPL FREE ワークエリア
\$0FF78	FF78	IPL RAM ジャンプ ワークエリア
\$0FF80	FF80	IPL カーソルX
\$0FF81	FF81	IPL カーソルY
\$0FF82	FF82	IPL ポジションY
\$0FF83	FF83	IPL カセットチェックサム(2バイト)
\$0FF85	FF85	IPL キーバッファ(1バイト)
\$0FF86	FF86	IPL PRINT OUT カラー
\$0FF87 IPLDRV	FF87	ディスクドライブNo.
\$0FF88	FF88	ディスクドライブスタックワークエリア(下位, 上位)
\$0FF8A	FF8A	ディスクエラーリターンアドレス(下位, 上位)
DSKTYP	FF8C	ディスクドライブタイプ 0.....2D 1.....2DD 2.....2HD 3.....* 2HD 4.....2D 5.....* 2D 6.....IS 7.....HD
\$0FFFE	FFFE	IPL #キー RAM ジャンプワークエリア

索引

8255..... 197

A

A/D変換..... 171
AM変調..... 246
AY-3-8910..... 235

B

BASIC用スタック..... 94
BIOSROM..... 88
BIOSルーチンの呼び出し..... 89
BUSY..... 265
Bresenhanのアルゴリズム..... 31

C

CGROM..... 149
CMTモデル..... 21
CPUアクセスバンク選択..... 111
CRC..... 207
CRTC..... 126
CRTCアクセス..... 128
CRTCイニシャライズ..... 128
CRTC設定..... 131
CRT画面構成..... 127
CTC..... 220

D

D/Aコンバータ..... 242
DCE..... 205
DMA..... 214, 233

E

EIA..... 205, 268

F

FAC..... 94
FAT..... 225
FDC..... 226
FDD..... 231
FIFOメモリ..... 171
FM変調..... 246
FM変調波..... 243

H

HDLC..... 206
HSVモデル..... 21
HuBASICのワークエリア..... 95

I

I/Oドライバ..... 88
I/Oポート..... 9
INT端子..... 202
IOCS..... 93

IPLROM..... 86

IPLリセット..... 86

IPLルーチンの呼び出し..... 88

IRQ割り込み..... 258

L

LFO..... 246

M

MERGEコマンド..... 35

N

NTSC信号..... 16, 167

O

OPM..... 242

P

PCGRAM..... 150
PCG外字方式..... 143
PCGキャラクタ方式..... 143
PPI..... 197, 235, 264

R

RGBモデル..... 21
RS-232C..... 205

S

SDLC..... 206
STROBE..... 265

U

USR関数..... 102

V

V-RAM構成..... 11
V-RAMモードの設定..... 39

X

X1turboシリーズの画面構成..... 129
X1の画面構成..... 128
X1フォーマット..... 223

Y

YM2149..... 242
YM2151..... 242
YM3012..... 242

ア

アタック部..... 252
アドレスバス..... 10
アドレスマークフラグ..... 229
アドレスレジスタ..... 126
アナグリフ方式..... 44
アトリビュートV-RAM..... 108

- アルゴリズム.....54, 243
 アンダーライン表示機能.....163
イ
 1次フォーマット.....223
 色.....21
 インターフェイス基板.....92
 インターラプト.....201
 インターレーススーパーインポーズ.....168, 271
ウ
 ウィンドウ.....18, 19
エ
 液晶シャッター.....46
 演算誤差.....102
 エントリーアドレス.....182
 エンベロープ.....56, 240
 エンベロープ形状設定レジスタ.....240
 エンベロープジェネレータ.....235
 エンベロープ周期設定レジスタ.....240
オ
 オペレータ.....54, 243
 音程.....52
 音量.....53
カ
 カーソル表示制御.....126
 解像度の設定.....38
 階調.....157
 階調数.....26
 外部拡張グラフィックパレットメモリ.....156
 外部クロック.....269
 外部システムプログラム.....87
 外部デバイス.....87
 カウンターモード.....220
 書き込みレジスタ.....215
 拡張パレット機能.....156
 拡張I/Oスロット.....204
 加算混合.....27
 カセットテープのフォーマット.....189
 画像処理.....171
 画面管理用I/Oポート.....133
 画面座標系.....18
 画面表示バンク表示.....111
 カラーコレクタ機能.....271
 簡易型RS-232C.....270
 漢字ROM.....139, 140, 151
 漢字ROMアドレス.....151
 漢字テキストV-RAM.....107
キ
 キーコードの補正.....260
 キーボードCPU.....176
 機械語モニター.....87
 帰線期間.....144
 輝度信号.....168
 キャラクタジェネレータ.....139
 キャリア.....54, 56, 243
ク
 グローシェーディング.....33
 矩形波.....246
 グラフィックV-RAM.....11, 110
 グラフィックパレット.....156
 クリッピング.....38
 グローバル変数.....36
 クロマキーコントロール.....174
ケ
 ゲームキー.....179
 減算混合.....27
コ
 高速アクセスモード.....147
 高速グラフィックルーチン.....38
 黒色制御機能.....165
 誤差項.....32
 コマンドレジスタ.....226
 コンティニューアスモード.....214
 コントロールバス.....10
 コントロールポート.....208
 コントロールライン.....198
 コントロールワード.....198
 コンピュータコントロール端子.....270
サ
 再送シーケンス.....207
 彩度(SATURATION).....22
 サイン波.....246
 サブCPU.....80, 176
 3角形表示のアルゴリズム.....33
 3次元表示の方法.....43
 サンプル&ホールド.....239
シ
 シャープPWM方式.....190
 時間定数レジスタ.....221
 色差信号.....168
 色相(HUE).....22
 時刻データ.....192
 時刻の設定と読み出し.....191
 システムI/Oポート.....92
 システムクロック.....260
 システムソフトウェア.....87
 自動同期制御回路.....167
 時分割方式.....46
 周波数成分.....51, 52
 周辺機器類.....84
 ジョイスティック.....266

シリアルI/O	205
シングルアクセスモード.....	90
振幅.....	53
ス	
スーパーインポーズ.....	184, 270
スーパーインポーズ機能.....	167
スクリーンモードの設定.....	39
スクロールアウト.....	169
スクロールインアウト.....	169
スクロール機能.....	168
スケーリング.....	20, 250, 253
スタートビット.....	206
スタビライジング機能.....	271
ステータス.....	231
ステータスレジスタ.....	226
ステップレートフラグ.....	228
ステレオスコープ方式.....	44
ストップビット.....	206
ストリングデータエリア.....	94
ストリングディスクリプタ.....	98
スペクトル.....	22
ズレ補正.....	173
セ	
セクタレジスタ.....	226
セントロニクスインターフェイス.....	263
ソ	
走査速度.....	16
増設RAM	185
タ	
タイマー.....	257
タイマーIC	191, 197
タイマーモード.....	220
タイリング.....	28
多色表示.....	120
多色モード.....	11, 129
多面体表示.....	33
チ	
チャンネル音量設定.....	238
チャンネルスイッチ.....	238
チャンネル制御レジスタ.....	221
中間言語.....	97
調歩同期式.....	206
直接アクセス.....	176
直線描画のアルゴリズム.....	31
ツ	
通常アクセスモード.....	145
ツリー構造.....	224
テ	
データセレクトIC	164
データベース.....	10

データポート.....	208
データレジスタ.....	226
テープフォーマット.....	190
ディケイ部.....	252
ディザリング.....	28
ディスクマップ.....	223
ディチューン.....	250
ディレイフラグ.....	229
ディレクトリ.....	224
テキストV-RAM	11, 106
テキストエンドコード.....	96
テキストパレット.....	161
デコード.....	10, 90
ディジタルテロップパー.....	270
デバイス座標系.....	18
テレビタイマー.....	194
テンポ設定.....	260
テンポラリストリングバッファ.....	94, 104
ト	
トーンジェネレータ.....	235
トーン周波数設定.....	237
同期.....	269
同時アクセスモード.....	90
ドキュメント化.....	35
トラック照合フラグ.....	228
トラックレジスタ.....	226
トラックレジスタ更新フラグ.....	228
トランスペアレントモード.....	214
取り込み開始コントロール.....	172
ナ	
内部拡張グラフィックパレットメモリ.....	156
内部クロック.....	269
内部レジスタ.....	126
ニ	
入出力命令.....	90
ネ	
音色.....	51
ノ	
ノイズ.....	246, 257
ノイズジェネレータ.....	235
ノイズ周波数.....	239
ノコギリ波.....	246
ノンインターレース.....	16
ハ	
バーストモード.....	214
倍音成分.....	51
バイトモード.....	214
バイナリモード.....	207
配列変数.....	99
バリディ.....	206

パレット機能	154
パレット設定用I/Oポート	155
パレットの初期化	39
パレットモード	11
反転機能	174
ハンドシェイク	265
汎用グラフィックルーチン	38
ヒ	
比較フラグ	229
日付データ	192
ピッチデータ	243
ビットセット、リセット機能	198
ビデオエンハンサー機能	271
ビデオ信号分解	171
ビデオデータの取り込み	171
ビデオマルチプロセッサ	271
非同期	269
ビブラート	248
ビューポート	19
表示開始アドレス	126
フ	
ファイル用ストリングバッファ	94
ファンクションコード	179
フィードバック	54, 252
フィールド	17
フォントのアクセス	144
物理アドレス	224
物理フォーマット	223
浮動小数点	101
ブランキング期間	165
プリセットブルタイマー	257
プリンタインターフェース	263
フレーム	17, 207
プログラマブル・キャラクタ・ジェネレータ	142
プログラムカウンタ	202
ブロッキングセーブ	190
ブロックIF	35
フロッピーディスクコントローラ	226
プライオリティ機能	164
ヘ	
ページの切り換え	136
ベースレジスタ	217
ベクタアドレス	182
ヘッドロードフラグ	228
変換マトリクス	20
偏光板方式	45
変数エリア	93
変数テーブル	97
ホ	
ボーレート	206

ホワイトノイズ	239, 252
マ	267
マウス	267
マウスの制御	213
マルチプレクサIC	164
マルチレコードフラグ	229
ミ	
ミュート	271
明度(VALUE)	22
モ	
モードA信号	179
モードB信号	179
モザイクコントロール	173
モジュール結合	35
文字ビット	206
モジュール変数	36
モジュレータ	54, 56, 243
モデム	270
ユ	
ユーザーI/Oポート	92
ユーザー座標系	19
優先順位	164
ヨ	
読み出しレジスタ	219
予約語	97
ラ	
ラインメモリ	171
ラッチ	164
リ	
リモコン信号	184
両眼視差方式	43
量子化コントロール	172
ル	
ルートディレクトリ	224
ルックアップテーブル	11
レ	
レイトレーシング	18
連続セーブ	190
ロ	
ローカル変数	36
論理アドレス	224
ワ	
ワールド座標系	19
割り込みデバイス	204
割り込みベクタ	201

●主な参考文献

I 部

- ・ **実践コンピュータグラフィックス**
D.F.ROGERS 日刊工業新聞社
- ・ **対話型コンピュータグラフィックス**
W.M.NEWMAN, R.F.SPROULL マグロウヒルブック
- ・ **DX SOUND MAKING BOOK**
小野峰人編 立東社
- ・ **FM音源スーパーサウンド**
窪田弘, 田中寿郎, 広瀬真 秀和システムトレーディング

II 部

- ・ **テクニカルマニュアル**
シャープソフト開発部監修 AZ ビジコム
- ・ **X1 リファレンスノウト**
杉浦勇一, 仲谷和人, 松村守, 難波生 エム・アイ・エー
- ・ **X1 シリーズ活用研究**
I/O編集部 工学社
- ・ **X1 turbo解析マニュアル**
伊牟田薫 秀和システムトレーディング
- ・ **X1 活用研究**
月刊マイコン編集部 電波新聞社
- ・ **X1 システム研究室**
有田隆也, 牛島昌和 日本ソフトバンク
- ・ **X1 110 番**
シャープソフト開発部監修 ラジオ技術社
- ・ **続・X1 110 番**
シャープテレビ第4商品企画部監修 ラジオ技術社
- ・ **oh./ MZ**
昭和61年4月号～昭和62年6月号 日本ソフトバンク

ご注意

- (1) 本書の一部又は全部について個人で使用するほかは、著作権上、(株)ビー・エヌ・エヌの承諾を得ずに無断で複写、複製することは禁じられております。
 - (2) 本書についての電話によるお問合せには一切応じられません。質問等がございましたら、往復はがき又は切手・返信用封筒を同封の上、弊社までお送り下さるようお願いいたします。
 - (3) 内容に関する責任は(株)ビー・エヌ・エヌにありますので、内容に関してメーカー等に直接問合わせすることは御遠慮下さい。
- 乱丁、落丁本はご面倒ですが弊社営業宛に御送付下さい。送料弊社負担にてお取替いたします。

X1 Techknow

定価 3,900円

発 行 1987年 8月 1日初版発行

著 者 ビー・エヌ・エヌ第2企画部編

発 行 人 樺島正博

発 行 所 株式会社ビー・エヌ・エヌ
〒102 東京都千代田区麹町4-5 紀尾井町レジデンス5F
電話 営業部：03-238-1321 編集部：03-238-1322

装 幀 ナチュラ

C T S 福田工芸

印 刷 東京音楽図書

製 本 豊栄製本

© BNN corp. 1987 Printed in Japan

ISBN4-89369-027-2 C3055 ¥3900E

X1-Techknow

X1 シリーズ

X1-turbo シリーズ

X1-turboZ

定価 3,900 円